Face Alignment in the Wild

Heng Yang

School of Electronic Engineering and Computer Science

Queen Mary University of London

Submitted in partial fulfillment of the requirements of the Degree of

Doctor of Philosophy

October 2015

To the memory of my beloved grandmother.

Declaration

I, Heng Yang, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below. I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material. I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis. I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university. The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Heng Yang
October 2015

Details of collaboration and publications related to this thesis:

- **Heng Yang**, Ioannis Patras, “Mirror, mirror on the wall, tell me, is the error small?" *CVPR*, Boston, MA, 2015.
- **Heng Yang**, X. He, X. Jia and Ioannis Patras,“Robust face alignment under occlusion via regional predictive power estimation." *IEEE Trans. Image Processing,* 2015.
- **Heng Yang**, X. Jia, Ioannis Patras, K. Pan, “Random subspace supervised descent method for computer vision problems". *IEEE Signal Processing Letters,* 2015.
- **Heng Yang**, Ioannis Patras,“Privileged information based conditional structured output regression forests for facial points detection."*IEEE Trans. Circuits and Systems for Video Technology,* 2015.
- **Heng Yang**, Ioannis Patras,“Fine-tuning regression forests votes for object alignment in the wild." *IEEE Trans. Image Processing*. December, 2014.
- **Heng Yang**, C. Zou, Ioannis Patras,“Face sketch landmarks localization in the wild." *IEEE Signal Processing Letters*, 2014.
- **Heng Yang**, C. Zou, Ioannis Patras, “Cascade of forests for face alignment." *IET Computer Vision*, 2014.
- X. Jia, **Heng Yang**, Kwok-Ping Chan, Ioannis Patras, “Structured semi-supervised learning for facial landmarks localization with face mask reasoning." *BMVC*, 2014.
- **Heng Yang**, Ioannis Patras,“Sieving regression forests votes for facial feature detection in the wild." *ICCV*, Sydney, Australia, 2013.
- **Heng Yang**, Ioannis Patras,“Privileged information-based conditional regression forests for facial feature detection." *IEEE FG*, Shanghai, China, 2013.
- **Heng Yang**, Ioannis Patras,“Face parts localization using structured-output regression forests." *ACCV*, Daejeon, Korea, 2012.

Acknowledgements

Many people have supported me during my time as a PhD student, and I would like to thank them all. This dissertation would not have been possible without the guidance of my supervisor, Dr. Ioannis Patras. I thank him for giving me the right amount of freedom and supervision at the same time. His strong encouragement rescued several of our works, at the moment when I almost gave up. I would like to thank Prof. Shaogang Gong and Dr. Pengwei Hao for keeping me on track in my yearly reports. I am grateful to Dr. Xuming He who hosted me during my visit to NICTA. I would like to thank my masters supervisor Prof. Xiaolin Liu. She convinced me to start my journey to research in my final college year. Without her help and encouragement, I would have never started a PhD. I am also grateful to Prof. Ebroul Izquierdo for his support in several aspects. I would like to thank my funding body China Scholarship Council for the support of my PhD studies.

I would like to thank Prof. Nikos Paragios and Prof. Richard Bowden for acting as examiners for my dissertation.

I would like to thank all MMV members for creating a very nice atmosphere in the past 3 years, Sertan, Petar, Daria, Wenxuan, Fiona, Oya, Julie, Saverio, Navid, Yixian, Shenglan, Yoshiki, Ivan, Zongyi and many others.

I would like to thank Stephanie for her patient discussions and giving me inspiration for creative ideas. I also thank my friends in Queen Mary, Yun Zhou, Xiatian Zhu, Mingying Song, Yansha Deng and many others for their consistent encouragement. I would like to thank my collaborator Xuhui Jia for several successful remote collaboration and our friendship.

I would like to thank my family especially my parents for providing me an opportunity of education, rather than pushing me to do farm work at a very young age, which was an essential privilege over my contemporaries at primary school.

Finally I am very thankful to my wife Jing for her love and devotion since we first met at 1999.

Abstract

Face alignment on a face image is a crucial step in many computer vision applications such as face recognition, verification and facial expression recognition. In this thesis we present a collection of methods for face alignment in real-world scenarios where the acquisition of the face images cannot be controlled. We first investigate local based random regression forest methods that work in a voting fashion. We focus on building better quality random trees, first, by using privileged information and second, in contrast to using explicit shape models, by incorporating spatial shape constraints within the forests. We also propose a fine-tuning scheme that sieves and/or aggregates regression forest votes before accumulating them into the Hough space. We then investigate holistic methods and propose two schemes, namely the cascaded regression forests and the random subspace supervised descent method (RSSDM). The former uses a regression forest as the primitive regressor instead of random ferns and an intelligent initialization scheme. The RSSDM improves the accuracy and generalization capacity of the popular SDM by using several linear regressions in random subspaces. We also propose a Cascaded Pose Regression framework for face alignment in different modalities, that is RGB and sketch images, based on a sketch synthesis scheme. Finally, we introduce the concept of mirrorability which describes how an object alignment method behaves on mirror images in comparison to how it behaves on the original ones. We define a measure called mirror error to quantitatively analyse the mirrorability and show two applications, namely difficult samples selection and cascaded face alignment feedback that aids a re-initialisation scheme. The methods proposed in this thesis perform better or comparable to state of the art methods. We also demonstrate the generality by applying them on similar problems such as car alignment.

Table of contents

Table of contents xi

List of figures xv

List of tables xxv

1 Introduction 1

1.1 Motivation . . . 1
1.2 Related work . . . 4
1.2.1 Local based face alignment . . . 4
1.2.2 Holistic based face alignment . . . 8
1.2.3 Other methods . . . 12
1.3 Contributions . . . 12
1.4 Datasets . . . 14
1.5 Structure of the Thesis . . . 17

2 Local-based Face Alignment 19

2.1 Regression forests . . . 19
2.2 Privileged Information based Conditional Structured-Output Regression Forests 23
2.2.1 Privileged information-based tree induction . . . 23
2.2.2 Models at leaf nodes . . . 27
2.2.3 Inference . . . 29
2.2.4 Evaluation . . . 31
2.3 Fine-tuning Regression Forests Votes . . . 46
2.3.1 RF votes with latent variable . . . 46
2.3.2 RF votes sieving . . . 47
2.3.3 RF votes aggregating . . . 49
2.3.4 Landmark unreliability . . . 52

2.3.5 Experiments . . . 52
2.4 Summary . . . 65

3 Holistic based Face Alignment **67**
3.1 General Framework of Cascaded Pose Regression . . . 67
3.2 Cascaded Forests for Face Alignment . . . 70
3.2.1 CPR training . . . 70
3.2.2 Forest-based regressor . . . 70
3.2.3 Intelligent initialization . . . 73
3.2.4 Experiment Setting . . . 74
3.2.5 Results . . . 75
3.3 Random Subspace based Supervised Descent Method . . . 83
3.3.1 Problem definition . . . 83
3.3.2 Random subspace SDM . . . 85
3.3.3 Evaluation . . . 88
3.4 Summary . . . 94

4 Robust Face Alignment Under Occlusion **97**
4.1 Supervised Occlusion Modelling for Face Alignment . . . 98
4.1.1 Problem definition . . . 98
4.1.2 Training preparation . . . 98
4.1.3 Structured decision forests . . . 99
4.1.4 Face mask reasoning and landmark localisation . . . 102
4.1.5 Experiment . . . 103
4.2 Unsupervised Occlusion Modelling for Face Alignment . . . 107
4.2.1 Problem definition . . . 107
4.2.2 Method . . . 108
4.2.3 Implementation details . . . 114
4.2.4 Results . . . 115
4.3 Summary . . . 122

5 Face Alignment Mirrorability **123**
5.1 Introduction . . . 123
5.2 Related Work . . . 125
5.3 Mirrorability in Face Alignment . . . 126
5.3.1 Mirrorability concepts and definitions . . . 126
5.3.2 Experiments . . . 127

5.4 Mirrorability Applications 129
5.4.1 Difficult samples selection 129
5.4.2 Feedback on cascaded face alignment 132
5.5 Comprehensive comparison 134
5.6 Summary and Discussion 135

6 Conclusion and Future Work **137**
6.1 Conclusion 137
6.2 Future Work 138

References **141**

Appendix A Sketch Face Alignment **151**
A.1 Problem definition 151
A.2 Method 154
A.2.1 Joint training of cascaded pose regression 155
A.3 Evaluation 156
A.3.1 Dataset and implementation details 156
A.3.2 Results on FSW 157
A.3.3 Results on LFPW test images 158

List of figures

1.1 The Detection-Alignment-Recognition (DAR) pipeline. 2

1.2 Example results the algorithms proposed in this thesis. 3

1.3 Illustration of the CLM procedure. 5

1.4 The mixture-of-trees model from (Zhu and Ramanan, 2012). Red lines denote springs between pairs of parts. 7

1.5 Illustration of the AAM procedure. 9

1.6 Referencing shape-indexed features comparison. 11

1.7 The 68 points mark-up used for annotations in 300-W. 15

1.8 Face image (left) and its mask annotation (right). 16

2.1 A binary classification tree during training (left) and testing (right). During training a set of labelled data point $\{\mathbf{x}\}$ is used to optimize the parameters of the tree, i.e. $h(\mathbf{x},\theta)$. During testing a test input data $\mathbf{x}$ is pushed through each tree ($t = 1,...,T$) and tested by the trained classifier until reach the terminal leaf node where posterior $p_t(c|\mathbf{x})$ is stored. 20

2.2 Regression Forests: training data and tree training. (Left) Input data points are shown in dark circles and the associated ground truth in denoted by their position along the y coordinate. The input feature space here is one-dimensional. (Right) A binary regression tree. During training a set of labelled training points is used to optimize the parameters of the tree. In a regression tree the entropy of continuous density associated with different nodes decreases when going from the root towards the leafs. (This figure is from (Criminisi et al., 2011b)) 21

2.3 Regression Forests: testing and the ensemble model. 22

2.4 An illustration of our proposed learning stage. The idealized tree induction for Privileged Information based Regression Forest (PI-RF) and Regression Forest (RF) is shown on the left. The training patches are from face images with a large variety w.r.t. the Privileged Information (PI) (here the head pose). A classical RF attempts to guide patches that are located around the same facial point to the same leaf node. However, as the example shows, the visual features vary due to changes in the PI and therefore it is difficult to guide them to the same leaf. On the contrary, in the PI-RF framework, the best split-function at some random internal nodes (in red) is selected directly according to the PI. As such, patches stored at the leaves tend to have low variation both in PI and in displacement. The information gain IG_{y} at dark nodes is calculated based on the entropy H_{y}, defined in Section 2.20 while at the color nodes, the information gain IG_{y^+} is calculated based on the entropy H_{y^+}, defined in Section 2.10. At each leaf node, one (or more) *base* feature point is defined and tree models are learned. 24

2.5 Structured Output Regression. (a) shows manually defined sparse spatial relations of parts on face based on their physical locations. 20 selected face parts (dots) are displayed and their relations are represented by dark lines. The purple dot is one representative facial point and its neighbouring points are the red dots with purple shadow. (b1) illustrates an example of the independence assumption between points used in previous regression forests methods. Here we use x_i to represent the voting element x that is able to vote for part i, i.e. x arrive at leaves of which the i-th point is the base point. (b2) shows the spatial shape model of our method, in which the position of the 4^{th} point does not only depend on its voting patches x_4 but also on the estimated positions of its neighbouring points in the structure graph. 32

2.6 An illustration of the structured output inference model. The face image shown here is *Laura_Flessel_0001.jpg* from LFW dataset. 32

2.7 Representative face images in BioID (left) and LFW (right) along with their facial point annotations. The green segments on the right face image represent our predefined graph model for the corresponding 10 facial points. . . 32

2.8 Conditional model vs. single model. Some representative results on LFW dataset. 35

2.9 Overall performance and comparison of RF and SORF on BioID dataset. . 36

2.10 Representative results from SO forests in LFW dataset, compared with their non-SO counterparts. 38

2.11 The performances of hybrid forests on LFW dataset, compared with the original ones. 39

2.12 CDFs of the m_{17} measure on BioID dataset, compared with reported results from (Cootes et al., 2012; Cristinacce and Cootes, 2008; Efraty et al., 2011; Milborrow and Nicolls, 2008; Rapp et al., 2011) 39

2.13 CDFs over point error on BioID dataset, compared with (Belhumeur et al., 2011; Cao et al., 2012; Martinez et al., 2012; Valstar et al., 2010). For fairness, only 17 internal facial points are used. 40

2.14 Overall performance of our method on LFW dataset, compared with (Dantone et al., 2012b). 40

2.15 Mean error of our model on LFPW dataset, compared to C-RF detector from (Dantone et al., 2012b). 43

2.16 Successful detection rate of our model on LFPW dataset, compared to C-RF detector from (Dantone et al., 2012b). 43

2.17 Example Images from LFPW dataset. First column shows detected facial points by C-RF (Dantone et al., 2012b), second column the detection results by PI-CRF-YAW forest and the last column the detected facial points by PI-CSORF-YAW Forest. 43

2.18 Comparison to RF based methods (C-RF (Dantone et al., 2012b) and RF-CLM (Cootes et al., 2012)) on AFLW. 44

2.19 Illustration of sieving via continuous latent variable (face center). (a) A voting element consists of two offset vectors, one to the target point (green arrow) and the other to face center (red arrow). (b) Original set of votes for the left brow center. (c) The absolute face center votes, those in green are regarded as consistent to the face center. (d) The remaining voting elements filtered by the face center sieve. (e) All voting elements are used to localize the face center (red dot). (f) and (h) are the Hough maps generated from votes of (b) and (d) respectively. (g) shows the corresponding detection results. 48

2.20 Illustration of aggregating the votes by updating the threshold. From left to right, the first row shows the original face image, all votes for the point ($\lambda = 0.35$), votes passed face center sieve and the aggregated votes from updated threshold ($\lambda = 0.22$) passed face center sieve. The color represents the weight of each vote and the dark terminal is the voting destination. The second row shows the detection results, normalized Hough map for original voting, after face center sieving and re-voting. 50
2.21 Feature extracted from the votes passed the face center sieve. The left shows an example image for training the one class SVM classifier for the left eye corder. The middle shows an example tested positive and the right shows an example tested as outlier. The red lines split the votes into four regions and the below shows their corresponding features, i.e., x_1. 52
2.22 Error distribution. 55
2.23 Face center estimation error distribution. 55
2.24 Relative improvement by using the face center sieve. 56
2.25 Performance of sieves associated with the face center on the AFLW. The left and right are landmark-wise mean error results on AFLW_TestI and AFLW_TestII respectively. Note that the Y axis range of (b) is different from that of (a). 56
2.26 Improvement plot over the baseline error (CRF-S). 58
2.27 Example results from the AFLW dataset before (top row) and after (bottom row) the votes aggregating. The value beside the red dot in the top row indicates the unreliability/ step length of aggregating. For clarity, the reliable point where no aggregating is needed, i.e. 0 is not shown in the figure. . . . 58
2.28 Detection results of example images from LFW. The upper shows the results by CRF detector (Dantone et al., 2012b) and the lower shows the results of our method. 59
2.29 Results on the LFW, compared to (Dantone et al., 2012b). The left and right are respectively the mean error decrease and accuracy increase on the LFW_TestI. 60
2.30 Results on the LFW, compared to (Dantone et al., 2012b).The left and right are respectively the mean error decrease and accuracy increase on LFW_TestII Note that the range of the Y axis is different from that of Fig. 2.29. 60
2.31 Results of our method on AFLW_TestI (Left) and AFLW_TestII (Right), compared to (Sun et al., 2012; Xiong and De la Torre, 2013; Zhu and Ramanan, 2012) and betaface.com (Betaface). 61

2.32 Results of our method on the AFLW, compared to random forests-based methods (Cootes et al., 2012; Yang and Patras, 2012). The numbers in legend of (c) are the percentage of test faces that have average error below 10%. 61
2.33 Left to right: Results for Mix.Tree, betaface.com, CRF-S, SO-RF and our method on an image from AFLW. The blue dots are the 12 common points. 63
2.34 An example image from AFW (Zhu and Ramanan, 2012) with results from Mix.Tree (Left) and our method (Right) 63
2.35 Landmark wise RMSE error for each view, from top to bottom: 1) all image, 2) images with no occlusions, 3) unoccluded landmarks of partially occluded image, 4) occluded landmarks of partially occluded image. 63
2.36 Comparison of the sorted RMSE for each view to the VCF model in (Boddeti et al., 2013), random forests model in (Li et al., 2011) and the baseline Regression Forests in our work. 64
2.37 Detection results of example images of different views from CMU-CW. The upper shows the results by plain regression forests and the lower shows the results of our method. 65

3.1 Starting from a raw pose, our method refines the face shape recursively by using different stages of regression forests, organised in a cascade. 71
3.2 Mean error of individual landmarks on the HELEN. 76
3.3 Performance against cascade levels on LFPW. 76
3.4 Performance against cascade levels on HELEN. 77
3.5 With different face detection initialization. 79
3.6 Example results of different methods on LFPW (the first row) and on HELEN (the second row). 82
3.7 RSSDM for face alignment. The image on the left shows the current pose. Then several subspaces are randomly generated, of which the cyan landmarks are selected and the red are not selected. Each regressor generates an update of the shape vector. The results are averaged as the final prediction of this iteration, as shown in the image on the right. 86
3.8 RSSDM performance with various number of subspaces and subspace dimensions. 90
3.9 RSSDM vs. SDM. The models are trained on training images in the HELEN dataset. The percentages in the legend show the proportion at the error level of 0.1. 91

3.10 RSSDM vs. SDM results with various Monte-Carlo numbers for model building. The models are trained on training images in the HELEN dataset. The figure on the left shows the results on Easy test set and the figure on the right shows the results on Challenging test set. 92

3.11 RSSDM vs. SDM results with various regularization parameters, where λ^* is the optimized regularization parameter. 92

3.12 3D human body and its image projection of the 3D points under a certain pose. 95

4.1 The images on the left side of the two pairs show the results from the standard Random Forests for facial landmarks localisation (Dantone et al., 2012b), with failure cases under occlusion. The images on the right side of the two pairs show the results of our proposed method. It first explicitly predicts the face mask (the semi-transparent region), then use the face mask information to improve the localisation and to predict the occlusion status of the landmarks. 99

4.2 The framework of proposed method. We use face images with annotation of facial landmarks and face masks for training. By randomly switching the information gain function at the internal nodes, the decision trees are optimized with respect to both the offsets to landmarks (regression) and to the local structured label configuration (classification). The forest model is able to predict the face mask and landmark locations jointly. We exploit the face mask prediction to further improve the landmark localisation. 100

4.3 Results on LFW_Test (a), LFPW_Test (b), and COFW (c), compared to (Betaface; Cootes et al., 2012; Zhu and Ramanan, 2012) and our previous methods from Section 2.2) and Section 2.3) . The error is measured as a fraction of the inter-ocular distance. LFW_Test and LFPW_Test only contain 'difficult' image. (d) shows the run-time performance in fps. 104

4.4 Illustration of two face mask reasoning results on COFW: (from left to right) original image, ground truth, result of the standard RF and result of our proposed method. 106

4.5 Illustration of the pipeline of the proposed method. Given a test image, we first detect the face and apply segmentation by the graph-based approach in (Felzenszwalb and Huttenlocher, 2004). Based on the face bounding box information and the segmentation result, we employ the local patch based Regression Forest voting method for face alignment and obtain the Regional Predictive Power map with pixel probability from α to 1. We then adapt the state of the art face alignment model, (Robust Cascade Pose Regression (RPP) is used as an example) by putting weights on different weak regressors. The final column shows the results from original RCPR (upper) and the adapted RCPR (lower). Our method is able to localise the landmarks more accurately (especially when occlusion is presented) and reason the occlusion labels of the landmarks (green = unoccluded, red = occluded). 108

4.6 Regression Forests (RF) voting based Region Predictive Power (RPP) estimation. (a) shows the original votes distribution inside the face bounding box, similar dense for both the face region and occlusion region. (b) shows the distribution after the face center sieving as in section 2.3. As can be seen, many invalid votes from the non-face parts are effectively removed, which is a strong cue to predictive the RPP. (c) is the over-segmentation result. (d) shows the RPP map, i.e., the p_r in Eq. 4.8, calculated over each region of the segmentation. (e) is the detection result from the local RF model with the color varies according to the reliability of the landmark estimation, described in Section 4.2.2. 110

4.7 The distribution of x_r^1 feature (a) and landmark reliability s_l (b) for facial regions and non-facial regions. In (b) the value s_l of one face is normalized in the range between 0 and 1. 115

4.8 Results on COFW, compared to CPR-family approaches (Burgos-Artizzu et al., 2013; Cao et al., 2012). 116

4.9 Comparison to the recent methods, SDM (Xiong and De la Torre, 2013), RCPR (Burgos-Artizzu et al., 2013), RF_Sieving in section 2.3, method of Yu et al. (Yu et al., 2013a), DRMF (Asthana et al., 2013), and CSRIO SDK (Cox et al., 2013) on COFW test images for their common 16 facial landmarks. For the DRMF, the pre-computed face bounding box model is used since the tree-based method does not work on such images. 117

4.10 Example results based on Viola-Jones face detector (blue) and 300-W face detector (red). SDM is trained based on Viola-Jones face detection and the other two are trained on 300-W face detection. The number under each pair shows increase of failure cases when face detection changes from one to the other. 118

4.11 Example results from COFW (first two rows) and LFPW and HELEN (last two rows), including landmarks detection results (upper) and the corresponding RPP map (lower). See Fig. 4.5 for color map definition. 120

5.1 Example pairs of localisation results on original (left) and mirror (right) images. The first column (a) shows large mirror error and the second (b) small mirror error. Can we evaluate the performance without knowing the ground truth? 124

5.2 Mirror error and alignment error of RCPR (Burgos-Artizzu et al., 2013) on 300W test images. Results are calculated over 68 facial points. 128

5.3 Mirror error and alignment error of GN-DPM (Tzimiropoulos and Pantic, 2014) on 300W test images. Results are calculated over 49 inner facial points. 129

5.4 Correlation between the alignment error and the mirror error of various state of the art face alignment methods. The correlation coefficients are shown above the figures. 130

5.5 Consistency measure of 'difficult' samples detection, with $M = 150$. 131

5.6 Restart scheme of our method vs. RCPR (Burgos-Artizzu et al., 2013) (best viewed in color). 134

5.7 Comprehensive comparison of our proposed holistic algorithms on 300w dataset. 135

A.1 Our approach trains a Cascaded Pose Regression model based on RGB face images and their synthesis (left), then estimates the facial landmarks locations in both face photos and face sketch images (right). 153

A.2 An example image of face sketch synthesis. From left to right are the original RGB image, edge detection by (Dollár and Zitnick, 2013) and our synthesized sketch image. In the synthesized image the eye regions and mouth region are enhanced and fused with the edge detection. 153

A.3 FSW example results. Face sketch images in FSW show large variety of head pose and drawing styles. The last image in the second row shows the average FSW individual landmark error levels, represented by the point sizes. 159

A.4 Results on the FSW dataset. The left shows the landmark-wise average error. The right shows the overall mean error. The landmark ID number definition please refer to (Sagonas et al., 2013c). Roughly, from #1 to #17 are landmarks along the face contour while the remaining are inner facial landmarks. For DRMF and OMP method, the inner mouth corners are not detected and their errors are shown as the mean value of all the landmarks. . 159
A.5 Results on the test images (RGB) of LFPW dataset. The figure configuration is the same to Fig. A.4. 160

List of tables

1.1 Description of datasets used in this thesis. 15

2.1 Mean error of each facial point in LFW dataset (%). 34
2.2 Successful detection rate of each facial point in LFW dataset (%). 34
2.3 Comparison of Mean Error (ME) and Successful Detection Rate (SDR) of forests that using and not head pose yaw privileged information (%). 37
2.4 Estimation accuracy of privileged information. 37
2.5 Percentages of test images with RMSE(Root Mean Square Error) less than the given thresholds on the LFW dataset, compared to (Cao et al., 2012; Liang et al., 2008) on LFW87 dataset. 42
2.6 SDM (Xiong and De la Torre, 2013) vs. our method when face BB shifts. . 45
2.7 Description of forest models trained on the AFLW 53
2.8 Aggregating steps proportion 57

3.1 Intelligent initialization vs. blind initialization. 77
3.2 Comparison to RF methods on LFW. 80
3.3 Comparison with the existing methods. C. represents the common 49 facial landmarks that SDM and other methods can detect while 66P represents the 66 common landmarks the methods except SDM can detect. 80
3.4 300-W dataset (68 landmarks). 93
3.5 300-W dataset (49 landmarks). 93
3.6 Rotation (in degree) and translation (in mm) errors of 3D body pose estimation. 94

4.1 Face mask reasoning results on the COFW dataset, compared to the related methods. 106
4.2 300-W dataset (68 landmarks). 118
4.3 300-W dataset (49 landmarks). 118

5.1 49/68 facial landmark mean error comparison 134

Nomenclature

λ	regularization parameter.
$\mathbf{S}$	set representation
$\mathbf{x}$	general data point in vector representation.
$\mathcal{H}(\cdot)$	entropy function.
$\mathcal{P}$	a set of image patches.
$\mathcal{X}$	general input space.
$\mathcal{Y}$	general output space.
$\mathrm{p}\rightarrow\mathrm{q}\mathrm{S}$	Mirror transformation of ${}^{p}S$ to the original image
$\mathrm{K} \in \mathfrak{R}^{3\times 3}$	camera intrinsic parameters.
$\mathrm{M} \in \mathfrak{R}^{3\times p}$	3D point vector.
$\mathrm{U} \in \mathfrak{R}^{2\times p}$	2D projection vector.
y^{+}	privileged information.
y_{i}	2D image position of landmark i.
${}^{p}S$	shape estimation on the mirror image
${}^{q}S$	shape estimation on the original image
e_{m}	mirror error in object alignment.
e_{a}	sample-wise alignment error in object alignment.
$h(\cdot)$	a feature extraction function.

I 2D image.

IG_{y} information gain calculated over y.

S shape vector representation of facial landmarks.

$Score(\cdot)$ scores accumulated from forests voting.

T number of trees in the forest.

X a set of input data points.

x a general point in input space, e.g. an image patch.

y a general output representation.

Z latent space.

Chapter 1

Introduction

1.1 Motivation

It is a common saying that *One Picture is Worth a Thousand Words*. Nowadays, with the rapid development of affordable high quality cameras and mobile phones, people take photos almost every day to record moments and memories. It turns out over 1.8bn photos are shared per day on Facebook, WhatsApp, Snapchat and Instagram alone in 2014. It provides a modern way of communication among family members, friends and even strangers. The human face is indisputably the most frequently appearing object in the photos, containing rich informations such as subject's identity, gender, emotion, age, hair style, ethnicity, kinship, etc.

Machine face analysis is an active research area in computer vision and artificial intelligence that is driven by a variety of real-world applications such as auto personal album organising, video surveillance, authentication, bio-metrics, computer animation, human-computer interaction, etc. For example, most of the current online social networks like Google+ and Facebook, that host a vast quantity of user photos, apply advanced face analysis techniques for automatic identity tagging, face annotation and face retrieval.

In such applications, taking face recognition as a concrete example, the processing usually follows the pipeline shown in Fig 1.1 (Huang, 2012). Given an image, first, face detection is carried out to localise faces in the image in question. The face detection result, usually in the form of a bounding box, is fed to the second step, face alignment, which aims at localizing a set of predefined facial landmarks, e.g., the eye corners, the nose tip and the mouth corners automatically. Then the face is aligned and fed to high level face analysis e.g., face recognition. Each step in this pipeline is essential as it has significant impact on the subsequent steps. As a result, a face recognition system is very likely to fail if the face alignment is incorrect or not sufficiently accurate.

Each separate stage of the pipeline has been intensively studied and much progresses have been made in the last decades. For example face detection technology has been embedded in most smart phones and cameras nowadays. This thesis focuses on the second stage, face alignment. There are several other problems that are very closely related or identical to face alignment in literature such as face registration (finding correspondences between face images), facial feature detection, facial landmarks localisation. Besides face recognition and face verification, there are several applications that demand reliable face alignment such as facial expression analysis (Moore and Bowden, 2011), face tracking, head pose estimation, gaze estimation, avatar animation, etc. Conceptually and technically, the face alignment problem is also very similar to other object alignment problems like Human Pose Estimation, bird part localisation, which involve localizing a set of predefined landmarks in images.

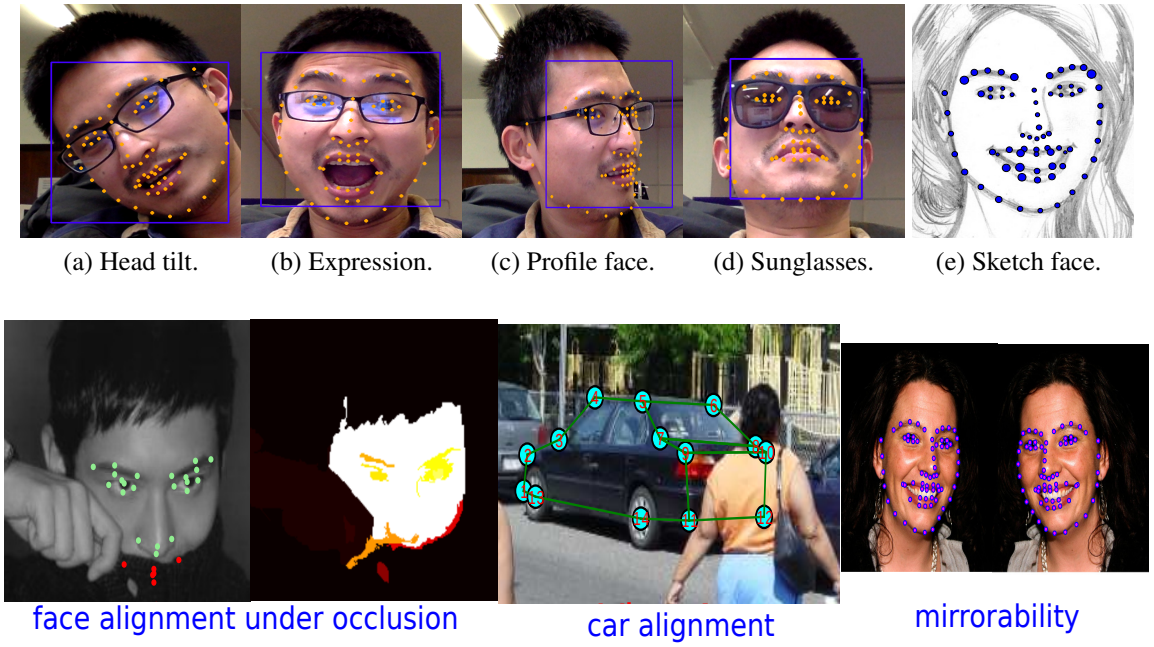
Fig. 1.1 The Detection-Alignment-Recognition (DAR) pipeline.

Due to its high demand, face alignment has been widely studied since 1960's when Woody Bledsoe, Helen Chan Wolf, and Charles Bisson created the first semi-automated facial recognition program. It was a man-machine system as it required the user to manually locate features such as the eyes, ears, nose, and mouth on the photograph (Bledsoe and Chan, 1965). Since then, researchers in this field have made great efforts to make a fully automatic and reliable face recognition system. One of the bottlenecks is an accurate face alignment. As (Bledsoe, 1964) pointed out:

> *This recognition problem is made difficult by the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc.*
> –Woody Bledsoe, 1966.

Such difficulties mentioned above have attracted many researchers in this field who have attempted a large variety of methods to tackle the problems. However, face analysis on images collected in unconstrained environment is still very challenging. The variability in head pose, expression and ageing usually result in misalignment and consequently lead to failures in recognition. Moreover, partial occlusion is also a tough challenge when dealing with face images collected in the wild. Ekenel and Stiefelhagen (Ekenel and Stiefelhagen, 2009) have carried out several experiments and pointed out that, the main reason that face

recognition algorithms fail on partially occluded face images is due to erroneous face alignment. In the first row of Fig. 1.2, a few examples are shown to demonstrate the several main challenges for face alignment.



(a) Head tilt. (b) Expression. (c) Profile face. (d) Sunglasses. (e) Sketch face.

Fig. 1.2 Example results the algorithms proposed in this thesis.

In this thesis we focus on face alignment and tackle the main challenges like head pose variations, facial expressions, heavy partial occlusions. We propose a set of accurate and fast methods for face alignment.

We first focus on local based method using regression forests given its good performance in other real-time computer vision applications. As face alignment is usually affected by head pose, we improve the traditional regression forest by explicitly using head pose as a privileged information and learning conditional models (on head pose) at leaf nodes. Traditional regression forests usually cast votes in a completely independent way, which occasionally result in inconsistent estimation. In order to solve this problem, we study on how to incorporate structure information and voting consistency within the forests.

Local-based methods have several limitations for instance it usually requires a detection model for every individual landmark. Therefore when the number of landmarks is big, the computational cost is high and the model size is also big. We later investigate holistic cascaded methods and address several common problems in cascaded learning such as reliable regression at each cascade and how to avoid over-fitting.

We later concentrate on face alignment under partial occlusion. Partial occlusion is one

of the most challenging problems when a face alignment model is applied on real world face images.We address this problem by explicitly estimating the face mask, i.e. a mask that marks the pixels that belong to the face, on the basis that facial area and background area should contribute in a different way in face alignment process. We address this problem in both supervised and unsupervised ways.

Finally, in order to effectively evaluate whether a face alignment system succeeds or not in practical applications, we introduce the concept of mirrorability, that is, the ability of a model/algorithm to preserve the mirror symmetry when applied on an image and its mirrored version. We introduce a measure called mirror error to measure mirrorability qualitatively. We demonstrate that the mirror error provides a very fast and inexpensive measure of the alignment error.

Some example results of the proposed algorithms are shown in Fig. 1.2.

1.2 Related work

A large number of methods are proposed in the past decades for face alignment. Generally speaking, there are two different sources of information typically used for face alignment: face appearance (i.e., texture of the face image) and the shape information. Based on how the spatial shape information is used we categorize the methods into local-based methods and holistic-based regression methods. The methods in the former category usually rely on discriminative local detection and use explicit deformable shape models to regularize the local outputs while the methods in the latter category, directly regress the pose (the representation of the facial landmarks) in a holistic way, i.e. the shape and appearance are modelled together. We review the methods in the two categories separately in Section 1.2.1 and Section 1.2.2 respectively. Finally we present some other methods designed for specific situations in Section 1.2.3.

1.2.1 Local based face alignment

A representative method in this category is the Constrained Local Model (CLM) (Cristinacce and Cootes, 2006; Saragih and Goecke, 2007). The general process of CLM fitting is shown in Figure 1.3. Its model usually consists of two parts. One for local detection, which is sometimes called local experts and the other for spatial shape models. Local expert model describes how image around each facial landmark looks in terms of local intensity or color patterns. Shape model describes how face shape, that is the relative location of the face parts, varies. This captures variations such as such as wide forehead, narrow eyes, long

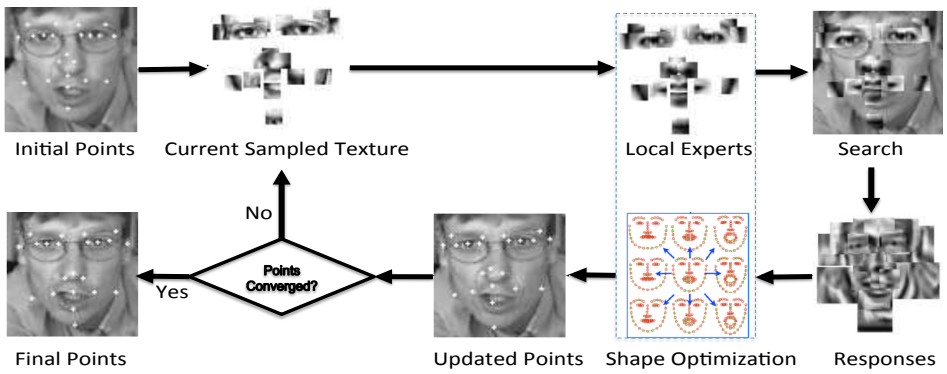


Fig. 1.3 Illustration of the CLM procedure.

nose etc.

The shape variation of the CLM can be represented by a linear model. The position of landmark i is represented by

$$\mathrm{y}_i = T(\bar{\mathrm{y}}_i + P_l b; t) \tag{1.1}$$

where $\bar{\mathrm{y}}_i$ is the mean position of the landmark in a suitable reference frame, P_i is a set of modes of variation, b are the shape model parameters and $T(\cdot;t)$ applies a global transformation with parameters t. To match the model to a new image I, one seeks the landmarks $\mathrm{y} = \{\mathrm{y}_i\}$ that optimise the fitting of the model to the image. Formally, the target is to seek parameters $p = \{b,t\}$ which minimise:

$$Q(p) = -\log(b,t|I) = -\log p(b) - \alpha \sum_{i=1}^{N} \log p(\mathrm{y}_i|I) \tag{1.2}$$

The scaling factor α is used to take account of the fact that the conditional probabilities for each landmark $p(\mathrm{y}_i|I)$ are not strictly independent. Assuming all poses are equally distributed thus $p(b,t) = p(b)$. Given an estimate of the scale and orientation, the quality of fit is calculated as $C_i(\mathrm{y}_) = -\log p_i(\mathrm{y}_|I)$, then the object function is

$$A(p) = -\log p(x) + \alpha \sum_{i=1}^{N} C_i(\mathrm{y}_i). \tag{1.3}$$

The first term encodes the shape prior while the second terms describes the image matching information. There are several methods in this framework. They differ from each other either in the way to model the shape prior or in the method for local detection. In what follows, we will first present the models for the shape prior then present local detection methods.

Shape Models: Early works used a multivariate Gaussian distribution to model shape. This is also known as Point Distribution Model (PDM) (Cootes and Taylor, 1995). Given a set of training samples with annotation of facial points locations, they first apply Procrustes analysis to remove the effect of rigid transformation. Then Principal Component Analysis (PCA) is applied to find the n largest eigenvectors. Since PCA can only capture linear variation of face shape structure, (Sozou et al., 1997) propose a new form of PDM, which uses a multi-layer perceptron to carry out non-linear principal component analysis. Romdhani, Gong and Psarrou modelled multi-view nonlinear active shape model using Kernel principal component analysis (KPCA). (De la Torre and Nguyen, 2008) presented an extension of KPCA for learning a non-linear appearance model invariant to rigid and/or non-rigid deformations. (Saragih et al., 2011) exploited the principal regression analysis to span a constrained subspace. CLM are widely adapted for instance (Asthana et al., 2013; Baltrušaitis et al., 2014; Cootes et al., 2012; Cristinacce and Cootes, 2006; Saragih and Goecke, 2007).

There are many other types of shape models that are different than PDM. One of the most successful models is the tree-structured model proposed by (Zhu and Ramanan, 2012). It consists of a mixture of trees that encode topological changes due to viewpoint as shown in Figure 1.4. As can be seen, there are no closed loops, maintaining the tree property. Unlike the densely-connected elastic graph models like (Wiskott et al., 1997), which is difficult to optimize, this model keeps the tree structure that can be globally optimized by dynamic programming. Each tree can be represented by the vertices V_m and the edges E_m as $T_m = \{V_m, E_m\}$, where m indicates a mixture and $V_m \in V$ is a set of landmarks. $\mathrm{y}_i = (x_i, y_i)$ is the pixel location of landmark i. Then the score of a configuration of landmarks $Y = \{\mathrm{y}_i : i \in V\}$ is given by:

$$Score(I, Y, m) = \mathrm{App}_m(I, Y) + \mathrm{Shape}_m(Y) + \alpha^m, \tag{1.4}$$

where

$$\mathrm{App}_m(I, Y) = \sum_{i \in V_m} \omega_i^m \cdot \phi(I, \mathrm{y}_i) \tag{1.5}$$

and

$$\mathrm{Shape}_m(Y) = \sum_{i,j \in E_m} a_{ij}^m d_x^2 + b_{ij}^m d_x + c_{ij}^m d_y^2 + d_{ij}^m d_y \tag{1.6}$$

The score of one configuration, i.e. Eq. 1.4, consists of the sum of the scores from shape and appearance, and a constant scale bias term associated with mixture m. The appearance term in Eq. 1.5 sums the appearance evidence for placing a template ω_i^m for landmark i, tuned for mixture m at location x_i. $\phi(I, \mathrm{x}_i)$ is the feature vector, a HOG descriptor (Dalal and Triggs, 2005), extracted around the pixel location x_i in image I. The shape term in Eq. 1.6 scores the mixture-specific spatial arrangement of landmarks X, where $d_x = x_i - x_j$ and

Fig. 1.4 The mixture-of-trees model from (Zhu and Ramanan, 2012). Red lines denote springs between pairs of parts.

$d_y = y_i - y_j$ are the displacement of the ith landmark and the jth landmark. Each term in the sum can be interpreted as a spring that introduces spatial constraints between a pair of parts, where the parameters (a,b,c,d) specify the rest location and rigidity of each spring. The inference of this model is to maximize this score term over Y and m:

$$Score^*(I) = \max_m[\max_Y Score(I,Y,m)] \tag{1.7}$$

This model has demonstrated good performance in capturing the global elastic face structures. It combines face detection, landmarks localisation and face pose estimation in the same framework which demonstrated good performance. This paper also proposed to share the template model of the facial landmarks across different views, which made the algorithm more efficient in terms of training the model. It has been further extended in (Yu et al., 2013a) for face detection and shape initialization. Ghiasi and Fowlkes (Ghiasi and Fowlkes, 2014) built a hierarchical model on top of the mixture of trees to deal with partial occlusions on face images.

Besides the parametric CLM model and the mixture-of-tree model, there are some non-parametric shape models. Belhumeur et al. (Belhumeur et al., 2011) proposed a RANSAC-like generate-and-test approach that constructs the shape constraints by iteratively selecting a optimal sub-set of the training images. An example of the so-called exemplars-based methods. (Zhou et al., 2013) proposed another exemplar based method using graph matching. (Smith et al., 2014) also utilized a similar RANSAC scheme for shape modelling, but their exemplars are directly from image and feature retrieval using an approximate nearest neighbour algorithm (Muja and Lowe, 2009). The Branch & Bound optimization technique was used in (Amberg and Vetter, 2011) for shape constraint modelling.

Local Experts: Local landmark (or part) detection is very similar to generic object detection, i.e., it treats each individual landmark as a specific object. Thus in this framework, many types of object detectors can be used for local landmark detection. The wide variety of local feature detectors that have been proposed can be broadly classified into classification-based, regression-based and voting based.

The classification-based approaches aim to design discriminative classifiers for each individual facial landmark based on the texture information of the specific landmark and its surrounding region. Different types of classifiers and image features are employed. For instance, in (Vukadinovic and Pantic, 2005), a GentleBoost classifier is proposed to detect each of the 20 facial points separately. The classic Support Vector Machine (SVM) classifier is used as facial point detector in (Rapp et al., 2011) and (Belhumeur et al., 2011). Various image features are utilised in the literature such as Gabor (Vukadinovic and Pantic, 2005), SIFT (Lowe, 2004; Xiong and De la Torre, 2013), HOG (Yan et al., 2013) and multichannel correlation filter responses (Galoogahi et al., 2013).

Regression-based approaches are also widely used. For instance, (Cristinacce and Cootes, 2007) presented a regression-based approach that combines a GentleBoost regressor with an Active Shape Model (ASM) that corrects the estimates obtained by the regressor. Another sequential regression-based approach was presented in (Martinez et al., 2012), where Support Vector Regressors (SVRs) were combined with a probabilistic MRF-based shape model. Apart from image features, geometry features are also used in the process of local detection. For example, line segments between facial points are also used to model the face shape in (Coşar and Çetin, 2011). Line segments are also used in (Efraty et al., 2011) as a type of geometry features for its cascade regression. A similar idea is employed in (Martinez et al., 2012) which goes one step further and considers the relations between any two line segments connecting two pairs of facial points.

Voting-based approaches accumulate votes for the position of the point given information in nearby regions. Since the introduction of the Generalised Hough Transform (Ballard, 1981) voting-based methods have been shown to be effective for locating shapes in images. The concept was extended for generic object detection. For instance, in the Implicit Shape Model (Leibe et al., 2004), local patches cast votes for the object position. Decision Forests for regression have been successfully applied to human pose estimation (Girshick et al., 2011) and facial feature detection (Dantone et al., 2012b). Image retrieval based voting approaches are proposed in (Shen et al., 2013; Smith et al., 2014). This line of methods first construct a large database of exemplar faces as well as their features (bag-of-words); then given a test image, they use a Hough voting scheme to retrieve the top exemplars for landmark voting; finally they use similar non-parametric shape regularization in (Belhumeur et al., 2011) to enforce shape constraints.

1.2.2 Holistic based face alignment

A typical method is in this category is the Active Appearance Model (AAM) (Cootes et al., 2001) which is generated by combing a model of shape variation with a model of the ap-

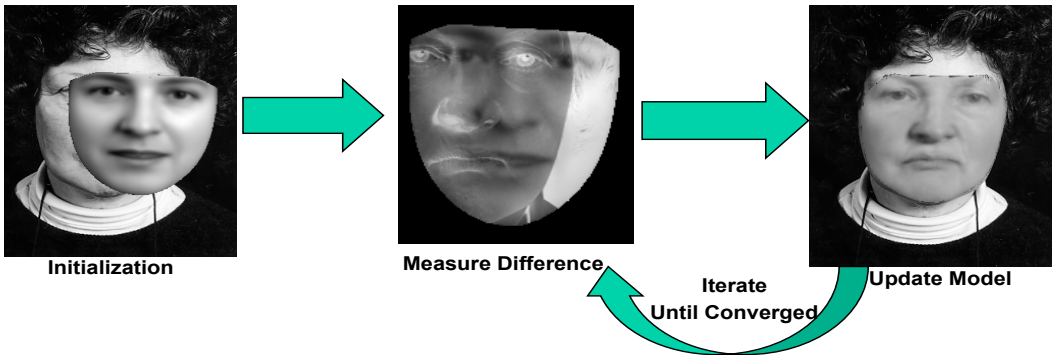


Fig. 1.5 Illustration of the AAM procedure.

pearance variations in a shape-normalised frame. Similarly to local based methods, first the shapes are pre-processed by Procrustes analysis. All of them are aligned into a common co-ordinate frame and the shape of each is represented by a shape vector S. PCA is applied on the data as a subsequent step. Then, any example can be approximated using:

$$S = \bar{S} + \mathrm{P}_s \mathrm{b}_s \tag{1.8}$$

where $\bar{S}$ is the mean shape, P_s is a set of orthogonal modes of variation and b_s is a set of shape parameters.

To build a statistical model of the grey-level appearance, first the example image is wrapped so that its control landmarks match the mean shape by triangulation as shown in Figure 1.5. To eliminate the effect of global lighting variation, the grey level of the wrapped sample g_m is normalized by a scale factor α and an offset β:

$$g = \frac{(g_m - \beta \mathbf{1})}{\alpha} \tag{1.9}$$

The values of α and β are chosen to best match the vector to the normalised mean. Then the texture model is also generated by applying PCA on all normalized textures as follows:

$$g = \bar{g} + \mathrm{P}_g \mathrm{b}_g \tag{1.10}$$

where $\bar{g}$ is the mean normalized gray-level vector, P_g is a set of orthogonal modes of variation and b_g is a set of grey-level parameters.

The shape and appearance of any example can thus be summarised by the vector b_s and b_g. Since there might be correlation between the shape and grey-level variations, then we can further apply PCA to the data and for each example we generate the concatenated vector

$$b = \begin{pmatrix} W_s \mathrm{b}_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s \mathrm{P}_s^T (\bar{S} - \bar{\bar{S}}) \\ \mathrm{P}_g^T (g - \bar{g}) \end{pmatrix} = \begin{pmatrix} Q_s \\ Q_g \end{pmatrix} c = Qc \tag{1.11}$$

where W_s is a diagonal weighting matrix determining the difference between the shape and texture parameters. The parameter vector c captures both the shape and texture variation, and Q are the eigenvectors.

Given a new image annotated with a set of landmarks at the mean pose, we can calculate the difference vector $\delta \mathrm{I}$ between the vector of grey-level values in the image I_i and the vector of grey-level values for the current model parameters, I_m:

$$\delta \mathrm{I} = \mathrm{I}_i - \mathrm{I}_m \tag{1.12}$$

We find the best match between the model and the image, by minimising the magnitude of the difference vector $\Delta = |\delta \mathrm{I}|^2$ by varying the model parameters, c. The original AAM method proposed a multivariate linear regression between the error in the model parameter and δI, thus:

$$\delta c = \mathrm{A} \delta \mathrm{I}. \tag{1.13}$$

AAM has been studied and improved in different aspects in the past decades. (Cootes et al., 2001) proposed a Gaussian-Newton optimization strategy.(Gross et al., 2005) suggest that the performance of an AAM built to model the variation in appearance of a single person across pose, illumination, and expression (a Person Specific AAM) is substantially better than the performance of an AAM built to model the variation in appearance of many faces, including unseen subjects not in the training set (a Generic AAM). (Saragih et al., 2008) applied a mixed inverse-compositional -forward-additive parameter update scheme to optimize the objective subject to soft correspondence constraints between the image and the model. (Amberg et al., 2009) proposed using the inverse compositional image alignment (ICIA) method for efficient and accurate AAM fitting. ICIA is extremely fast but has a small convergence radius. Therefore they proposed two novel fitting schemes, the compositional gradient descent and the linearised compositional descent. (Lucey et al., 2013) extended the inverse compositional method in the frequency Fourier domain for image alignment and applied this method for AAM fitting. (Tzimiropoulos et al., 2014) employs a statistically robust appearance model based on the principal components of image gradient orientations. They show that when incorporated within standard optimization frameworks for AAM learning and fitting, the kernel Principal Component Analysis results in robust algorithms for model fitting. (Tzimiropoulos and Pantic, 2013b) studied the problem AAM fitting on images collected in the wild.

Similar to the AAM is the cascaded pose regression (CPR) (Dollár et al., 2010). The pose is also represented as a set of landmark locations $\bar{S}$, which we call shape in this work. The shape is initialized by a mean shape $\bar{\bar{S}}$ or a random shape, which is updated iteratively. Though conceptually it is very close to the AAM, there are two main differences. First, the CPR uses pose-indexed features, instead of grey-level differences; Second, there is no convergence measure as no image difference computation is involved, instead, the method terminates after a fixed number of iterations. Noticeable progresses have been made in recent years in this framework. They mainly differ from each other in terms of the regression method that is applied at each iteration. For instance, in the original CPR method, a random regression fern was used as the primitive regressor. This was extended by (Cao et al., 2012), who used a two-level cascaded learning. (Cao et al., 2012) also introduced a fast correlation-based feature selection strategy for fast training. This framework was further improved by (Burgos-Artizzu et al., 2013). Instead of extracting image features with respect to the location of the closest facial landmarks, they used reference pixels in between two landmarks. This proved to be more effective in dealing with large head pose variations. The comparison of the feature extraction is shown in Figure 1.6. Besides this cascaded

(a) Feature in (Cao et al., 2012) (b) Feature in (Burgos-Artizzu et al., 2013)

Fig. 1.6 Referencing shape-indexed features comparison.

framework based on random ferns, there are other successful strategies. (Kazemi and Sullivan, 2014) replaced the random ferns with random forests as the primitive regressor. (Xiong and De la Torre, 2013) proposed a method that extracts SIFT features surrounding the landmarks at current shape, then uses a linear regression to calculate the update on the shape. It leads to fast convergence (usually less than 4 steps) and highly accurate face alignment. The SIFT feature was replaced by the HOG feature in an extension of their work which showed better performance (Xiong and De la Torre, 2014). (Asthana et al., 2014) proposed a different training strategy for this framework which enables incremental training. Instead of using hand-crafted features, (Ren et al., 2014) used features learned from random forests, which they call Local Binary Features. Recently, deep learning techniques are also applied in this problem, including convolutional networks (Sun et al., 2013; Zhang et al., 2014d)

and Auto-Encoder (Zhang et al., 2014b).

Most of the cascaded methods in this category depend on the initialization that is usually derived from the face bounding box. Current CPR based methods like (Burgos-Artizzu et al., 2013; Cao et al., 2012; Dollár et al., 2010) attempt to deal with this issue by initializing the method with several shapes and then by selecting the median value of the outputs. Burgos-Artizzu et al. (Burgos-Artizzu et al., 2013) proposes a *smart restart* scheme to improve the robustness to random initialization.

1.2.3 Other methods

There are some other methods that are designed to deal with specific issues. For example, (Pedersoli et al., 2014) studied the problem of localizing face and facial landmarks under weak supervision. They model the faces as a densely and uniformly distributed set of parts connected with pairwise connections, forming a graph. The immediate advantage of this representation compared to the tree of parts of (Zhu and Ramanan, 2012) is that the model does not need to know where the facial landmarks are, because parts are placed uniformly over the entire face. Once the face is detected using such a graph, the landmarks can be estimated given an example with landmark annotation. A semi-automatic methodology for facial landmarks localisation is proposed in (Sagonas et al., 2013b). An interactive way of facial landmarks localisation based on Active Shape Model in very high resolution images is proposed in (Le et al., 2012). In this interaction model, a user can efficiently guide the algorithm towards a precise solution. Since different databases for face alignment are annotated in different ways in terms of number of landmarks and the set of landmarks, (Zhu et al., 2014) and (Smith and Zhang, 2014) proposed methods of transferring the annotation across databases.

1.3 Contributions

The ultimate goal of this thesis is a robust and accurate face alignment system that works in uncontrolled environments at real-time speed. Given the accuracy and run-time performance of random regression forests on facial feature detection (Dantone et al., 2012a), we first focus on building regression forest models. As the original model only focuses on discriminative local detection, we address two main issues, head pose variation and noisy local appearance, for example partial occlusion and shadows. However, as a local based method, a regression forest model is inherently inefficient when the number of facial landmarks is large. We then focus on holistic models, i.e. the cascaded framework. We address several

problems in cascaded face alignment that include the fitting problem of regression in each update, the initialization problems. As we found heavy partial occlusion is the main challenge in face alignment in the wild, we focus on addressing this issue specifically by explicit facial region reasoning. Finally, we study the mirrorability of general object alignment. We also released several open source codes of our proposed methods. The main contributions of this thesis can be summarized as follows:

- We first improve the standard regression forest for local based face alignment. We focus on 1) learning higher quality decision trees using privileged information for local experts; 2) learning structure information within forests instead of using explicit shape models. The privileged information is only available at training time and it can be used both in selecting the split function at some randomly chosen nodes and in learning a conditional voting model at the leaf node. We model the shape constraints between the locations of the different points within the forest. In this way, the shape models are naturally conditioned on the test images as well as the privileged information.

- We propose a fine-tuning scheme that refines the regression forest votes for object alignment before accumulating them into the Hough space, by sieving and/or aggregating. We use a bank of sieves to filter out votes that are not consistent with some latent variable (e.g. the face center), something that implicitly enforces global constraints similar to shape models. In order to aggregate the votes when necessary, we adjust a proximity threshold at each iteration by applying a classifier on mid-level features extracted from voting maps for the object landmark in question.

- We propose three holistic based methods for face alignment. The first one, cascaded regression forests, uses regression forest as the primitive regression at each iteration in the cascade, which outperforms the conventional random ferns based method. The second one, random subspace based supervised descent method, is an extension of the supervised descent method by using several weak regressors in random subspaces. It maintains the high accuracy on training data and improves the generalization accuracy. The third one is a cascaded method specifically for facial landmarks localisation in multi-modality face images. It is based on a simple yet effective face sketch synthesis system and a joint training scheme. We introduce a data set called Face Sketches in the Wild (**FSW**), with 450 face sketch images collected from the Internet and with the manual annotation of 68 facial landmark locations on each face sketch.

- We propose two types of methods specifically for face alignment under heavy occlusion. The first one models patch occlusion status explicitly, in a way similar to seman-

tic image labelling, by encoding each pixel with a semantic label, face or non-face in our case. It then forms a structured semi-supervised forest framework for face mask reasoning and landmarks localisation. The second one is an unsupervised framework. It employs an over-segmentation method to partition the image into non-overlapping regions and predicts the power of each region, i.e., the Regional Predictive Power (RPP) and is essentially a measure of how useful information from a certain region can be for the task of face alignment. The output of this step is a dense RPP map that also indicates the probability of each region belonging to the face. This RPP map is then used along with the original face image for final face alignment using an adapted Cascaded Pose Regression methods. In order to evaluate the effectiveness, we extend the COFW dataset (Burgos-Artizzu et al., 2013) with face mask labelling in pixels.

- We finally propose a general concept in object part localisation, i.e. mirrorability which exploits whether object part localisation methods produce bilaterally symmetric results on mirror images. We introduce a corresponding measure, namely the *mirror error* to evaluate the mirrorability on two representative problems, namely human pose estimation and face alignment. Our experiments lead to several interesting findings for example the high correlation between the mirror error and the alignment error. We also show two valuable applications, difficult sample selection and feedback for cascaded face alignment, to show its usefulness

1.4 Datasets

In order to evaluate our proposed algorithms, we use the following datasets throughout this thesis. There are several datasets are collected for the problem of face alignment, including those collected in the laboratory in early years, such as BioID (Jesorsky et al., 2001), XM2VTS (Messer et al., 1999) and PUT (Kasinski et al., 2008) and those collected from the Internet such as LFPW, LFW. We list the publicly available datasets that we have used in our thesis as shown in Table 1.1. We make a brief review of the characteristics of all the datasets below.

The **BioID dataset** (Jesorsky et al., 2001) has been recorded in a laboratory environment using a low-cost web-cam. It consists of 1521 images, each depicting a frontal view of face of one of 23 different subjects with various facial expressions. One representative image from this dataset is shown in Fig. 2.7. Most of the previous methods in the topic of facial point detection have reported their results on this dataset. This allows us to compare our work with the state-of-the-art methods.

Table 1.1 Description of datasets used in this thesis.

Datasets	# landmarks	Resolution	# images	Used in	Notes
BioID	20	low	1521	Sec.2.2	Multiple images for one person
LFW	10	low	13233	Sec.2.2,2.3	Multiple images for one person
LFPW	29	high	871+239	Sec. 3.2	Only urls are provided
AFLW	up to 21	diverse	25993	Sec.2.2,2.3	Not fully annotated
300W	68	high	135+300	Sec. 3.2, 3.3,4.2, Chapt.5	Incl. LFPW, HELEN, AFW
COFW	29	diverse	1007	Sec.4.1, 4.1	Landmarks visibility annotated
COFWM	29	high	507	Sec.4.1, 4.2	With face mask annotation.
CMU-CW	8-14	low	4724	Sec.2.3	From 4 different views

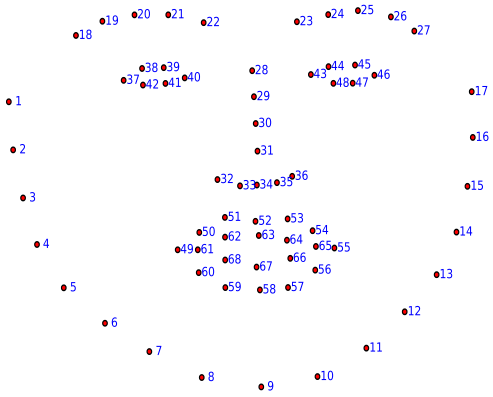


Fig. 1.7 The 68 points mark-up used for annotations in 300-W.

The **Labelled Face in the Wild (LFW) dataset** (Huang et al., 2007) has been designed for studying the problem of unconstrained face recognition. It contains more than 13,000 face images collected from the web. It consists of face images from 5749 individuals, 1680 of which have two or more distinct photos. (Dantone et al., 2012b) have annotated 13,233 faces for this dataset with the location of 10 facial points. The images exhibit a large variation in face appearances (e.g., pose, expression, ethnicity, age, gender) as well as general imaging and environmental conditions.

The **Labelled Face Parts in the Wild (LFPW)** is also a dataset with face images in the wild. The images are downloaded from the Internet under a variety of acquisition conditions, including large variability in pose, illumination, expression, partial-occlusion of the face. This dataset shares only image URLs on web but some of them are no longer valid. Around 800 of the 1132 training images and 220 of the 300 test images could be downloaded when we carried out the experiment.

The **Annotated Face Landmarks in the Wild (AFLW)** (Kostinger et al., 2011) contains real-world face images from Flickr. These images exhibit a very large variability in pose, lighting, expression as well as general imaging conditions. Many images exhibit partial occlusions that are caused by head pose, objects (e.g., glasses, scarf, mask), body parts (hair, hands) and shadows. We selected a subset in which all 19 frontal landmarks (i.e.

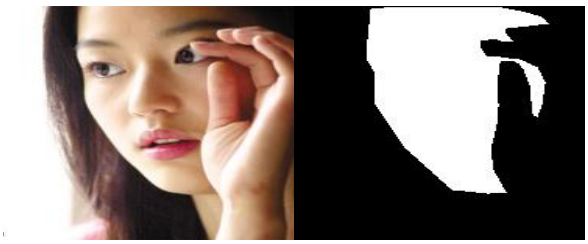


Fig. 1.8 Face image (left) and its mask annotation (right).

excluding the two ear lobes) were annotated that consists of 6200 images.

300W was created for Automatic Facial Landmark Detection in-the-Wild Challenge using a semi-automatic annotation methodology (Sagonas et al., 2013a). Landmark locations for several popular data sets including LFPW, AFW and HELEN are re-annotated with the same 68 points mark-up. In addition, it contains a new set called iBUG where the images are more challenging. It provides a good benchmark for face alignment evaluation. The 68-point-mark-up is illustrated in Fig. 1.7.

COFW: Caltech Occluded Faces in the Wild. This dataset (Burgos-Artizzu et al., 2013) consists of 1007 face images showing heavy occlusion and large shape variations, which was designed to benchmark face landmark algorithms in realistic conditions. All images were hand annotated with 29 landmarks and their corresponding visibility flag as well. Since they were obtained from a variety of sources, the faces are occluded by various patterns (e.g., hands, hats, hair, sunglasses, etc.) in different degrees. **COFW dataset with face mask (COFWM)** is an extension of the very challenging COFW dataset. We provide 496 images with face mask annotation for training and 507 images for testing. The face mask indicates whether a pixel inside a face image belongs to the face (1) or not (0). Some example images are shown in Fig. 1.8.

CMU Cars in Wild (CMU-CW) (Boddeti et al., 2013) contains 3433 cars spanning a wide variety of types, sizes, backgrounds and lighting conditions including partial occlusions. The images are from MIT Street Dataset created for the task of object recognition and scene understanding. The car landmarks were annotated in (Boddeti et al., 2013). The labelled data was manually classified into five different views: 932 frontal view, 1400 half-front view, 1230 half-back view and 1162 back view images. The car shape is respectively represented by 8, 14, 10, 14 and 8 landmarks.

1.5 Structure of the Thesis

Chapter 2 Local based Face Alignment In this chapter, we present two types of local based Regression Forests methods that improve face alignment performance. The first one, *Privileged Information based Conditional Structured Output Regression Forest* focuses on learning better structure of random trees by employing privileged information for local experts detection and introducing shape constraints within tree building. Part of this framework was originally presented in (Yang and Patras, 2012, 2013a) and (Yang and Patras, 2015). The second one, *Fine-tuning Regression Forests Votes*, focuses on analysing the local votes from regression forests in the testing process by sieving invalid votes and/or aggregating votes when necessary. The idea of vote sieving was originally presented in (Yang and Patras, 2013b). The overall vote fine-tuning framework was presented in (Yang and Patras, 2014). We carry out evaluation experiments of face alignment on the classic *BioID dataset* and datasets collected in the wild including the LFW, LFPW and AFLW. To demonstrate the generality of the proposed scheme, we show car alignment results on the CMU-CW dataset.

Chapter 3 Holistic based Face Alignment In this chapter, we present three holistic method for face alignment based on cascaded learning. In the first one, we introduce regression forests into the face alignment cascaded framework as the primitive regressor. This work was originally presented in (Yang et al., 2014). In the second one, we use random subspaces in the Supervised Descent Method (SDM) framework for better generalisation for both face alignment and 3D pose estimation. We presented this work in (Yang et al., 2015b).

Chapter 4 Robust Face Alignment under Occlusion In this chapter, we present two methods specifically for face alignment under heavy occlusion, which is the main challenge of face alignment in the wild. In the first one, with a subset of training images annotated with a face mask, we focus on explicit occlusion modelling and present a structured semi-supervised forest framework for simultaneous face alignment and occlusion reasoning. We originally presented this work in (Jia et al., 2014). In the second one, we model the occlusion in an unsupervised way. We exploit the voting consistency in each of the face regions given by image segmentation and obtain the *Regional Predictive Power* map. The RPP map is then utilized to adapt a cascaded model for consistent regression. This work was presented in (Yang et al., 2015a). We evaluated these two methods on the COFW dataset which we extended to include face mask annotation.

Chapter 5 Object Alignment Mirrorability Finally we study a general problem in object part localisation, i.e. the mirroability, defined as the ability of a model/algorithm to preserve the mirror symmetry when applied on an image and its mirror image. We use a measure, *mirror error* to evaluate the mirrorability of several algorithms in two representative problems, namely the face alignment and human pose estimation, across different methods and different datasets. We show interesting findings of mirroability and its usefulness in two applications.

Chapter 2

Local-based Face Alignment

In this chapter, we address the face alignment problem using local based methods. We will focus on how to use random Regression Forests for building a better local expert detector as well as for encoding shape constraints. We address several traditional challenges in face alignment in images collected in unconstrained environments namely head pose variations and partial occlusions. We first introduce the general framework of Random Forests, particularly the random Regression Forests in Section 2.1. In section 2.2, we present a privileged information based conditional structured-output regression forest which aims at building better decision trees for face alignment by using privileged information and by learning shape constraints with the forests. In section 2.3, we present a fine-tuning scheme that refines the regression forest votes (sieving and/or aggregating) before accumulating them into the Hough space for face alignment. The proposed methods are evaluated on the traditional face alignment datasets as well as datasets collected in the wild. Extended application on car alignment is also presented in this chapter.

2.1 Regression forests

Tree-based methods are well appreciated among practitioners because they can produce simple and easy to interpret rules relating an outcome to a set of covariates by recursively dividing the data into nodes that are as homogeneous as possible with respect to the outcome variable. Since Breiman (Breiman, 1984) introduced decision trees for classification and regression (CART) and Quinlan (Quinlan, 1993)'s "C4.5" algorithm for training optimal decision trees from data, decision trees have been widely used. In early works, single trees were used. Recently with the increased availability of computing resources and speed, it has emerged that using an ensemble of learners (e.g. weak classifiers) yields greater accuracy and generalization. The popular bagging ensemble algorithm works by aggregating many

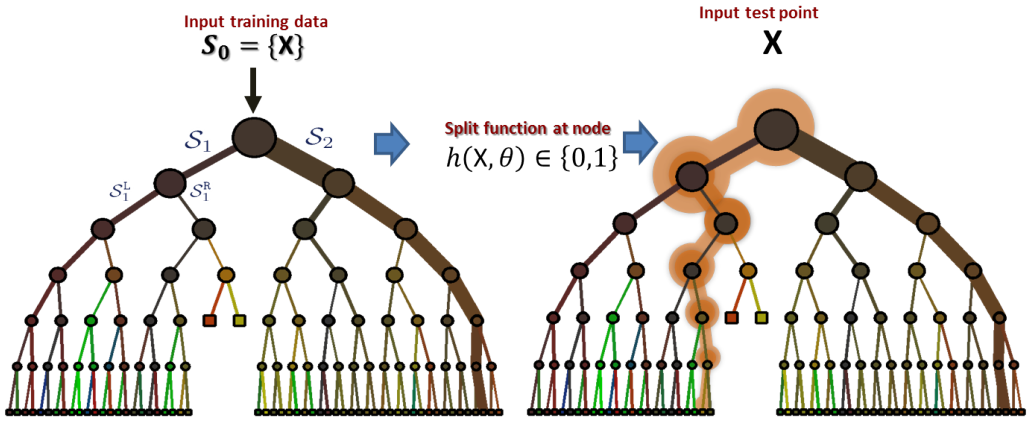


Fig. 2.1 A binary classification tree during training (left) and testing (right). During training a set of labelled data point $\{\mathbf{x}\}$ is used to optimize the parameters of the tree, i.e. $h(\mathbf{x}, \theta)$. During testing a test input data $\mathbf{x}$ is pushed through each tree ($t = 1, ..., T$) and tested by the trained classifier until reach the terminal leaf node where posterior $p_t(c|\mathbf{x})$ is stored.

trees, i.e., random decision forests. Recent years have seen an explosion of random forests-based techniques in machine learning and vision literature (Bosch et al., 2007; Fanelli et al., 2011), especially for human pose estimation using Microsoft Kinect (Shotton et al., 2013).

Previous research has focused on training and studying the factors that most influence the behaviour of a decision forest including the forest size, the maximum allowed tree depth, the amount of randomness and its type, the training objective function, etc. Shotton et al. (Girshick et al., 2011; Shotton et al., 2008) discussed how the testing accuracy increases monotonically with the forest size. As pointed out in (Shotton et al., 2013), very deep trees can lead to over-fitting although using very large amounts of training data mitigates this problem.

Recently, Criminisi et al. (Criminisi et al., 2011b) presented a unified model of decision forests which can be used to tackle all the common learning tasks: classification, regression, density estimation, manifold learning, semi-supervised learning and active learning. The training and testing of each tree is illustrated in Fig. 2.1. A data point is denoted by a vector $\mathbf{x} = (x_1, ..., x_f, ..., x_F) \in \mathbb{R}^F$ and the set of all points denoted by $\mathbf{S}$. Each split node j is associated with a binary split function

$$h(\mathbf{x}, \theta_j) \in \{0, 1\}, \tag{2.1}$$

which is the weak learner that is characterized by its parameters $\theta = (\phi, \psi, \tau)$ where ϕ randomly selects some features of choice out the the entire vector $\mathbf{x}$, e.g. $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, with $d' \ll d$, ψ defines the geometric primitive used to separate the data (e.g. an axis-aligned

hyperplane, a general surface etc.) and vector τ captures the thresholds for the inequalities used in the binary test. The main goal of training is to optimize all the parameters θ_j^* of the j^{th} split node. This is done in (Criminisi et al., 2011b) by maximizing an information gain objective function:

$$\theta_j^* = arg\max_{\theta_j} I_j \tag{2.2}$$

with

$$I_j = I(\mathbf{S}_j, \theta_j) = I(\mathbf{S}_j, \mathbf{S}_j^L(\theta_j), \mathbf{S}_j^R(\theta_j)). \tag{2.3}$$

The symbols $\mathbf{S}_j, \mathbf{S}_j^L, \mathbf{S}_j^R$ denote the sets of training points before and after the split expands. The objective function I_j varies from different application areas.

Regression forests

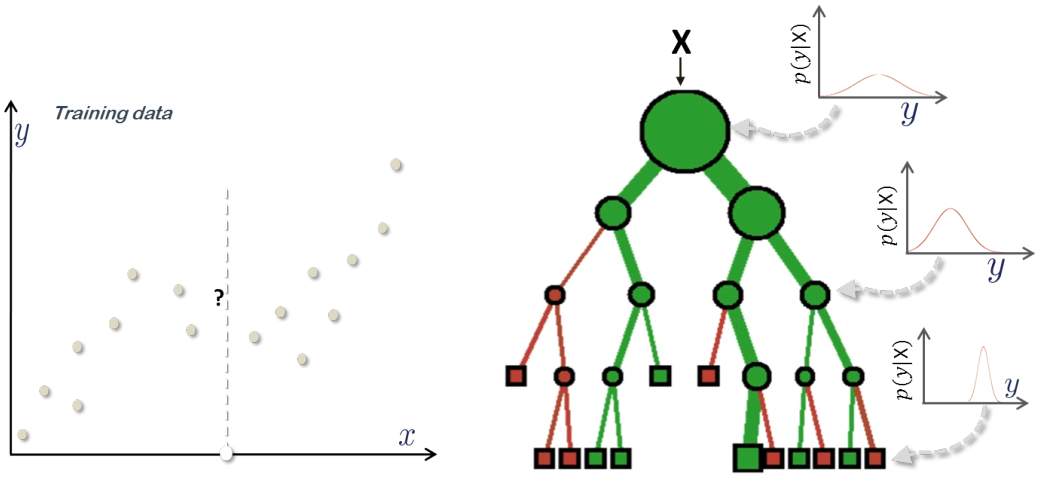


Fig. 2.2 Regression Forests: training data and tree training. (Left) Input data points are shown in dark circles and the associated ground truth in denoted by their position along the y coordinate. The input feature space here is one-dimensional. (Right) A binary regression tree. During training a set of labelled training points is used to optimize the parameters of the tree. In a regression tree the entropy of continuous density associated with different nodes decreases when going from the root towards the leafs. (This figure is from (Criminisi et al., 2011b))

Classification forests are very widely used, interesting authors are referred to (Criminisi et al., 2011b) for a comprehensive introduction. While the classification forest yields class probability at the leaf node, the regression forest predicts continuous output. The training

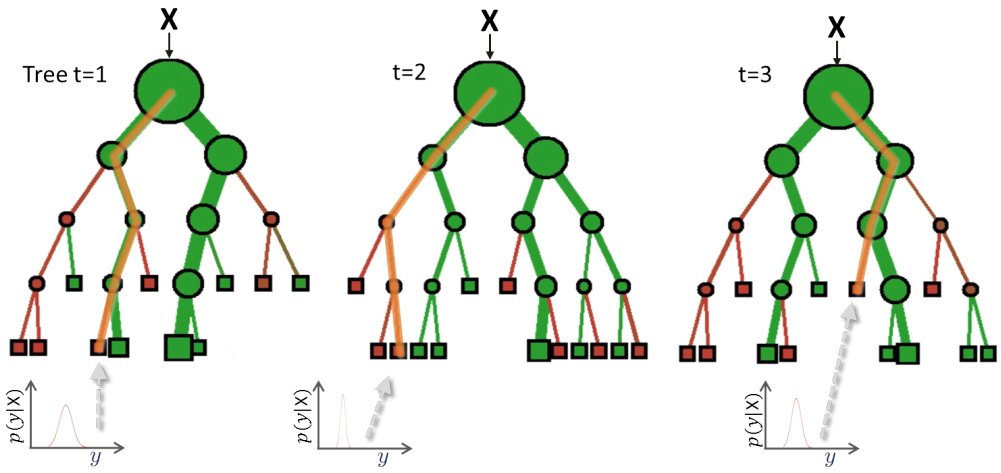


Fig. 2.3 Regression Forests: testing and the ensemble model.

process is shown in Fig. 2.2. The training labels are continuous. Consequently the objective function has to be adapted accordingly. The testing procedure is shown in Fig. 2.3. Each tree generates a continuous prediction and the regression forest posterior is simply the average of all individual tree posteriors:

$$p(y|\mathbf{x}) = \frac{1}{T}\sum_{t}^{T} p_t(y|\mathbf{x}). \tag{2.4}$$

Random Forest can be regarded as an important instance of the Generalized Hough Transform (Kontschieder et al., 2012). Hough Forests in (Gall et al., 2011) simultaneously cluster image features based on their spatial distribution and train a discriminative classifier using randomized decision trees. Associating a collection of 'vote offsets' with leaf node, the method then accumulate all discretized votes to determine the centres of the objects. In order to deal with a large dataset and estimate human pose from depth image, Girshick et al. (Girshick et al., 2011) use offset clustering through mean-shift at each node during training and a continuous voting space for testing. It has shown to provide good performance in object detection, tracking and action recognition and has achieved significant progresses in various challenging computer vision tasks. In (Fanelli et al., 2011), regression forests are used for head centre localisation and head pose estimation using depth images. It has been extended to regression forest in (Girshick et al., 2011) for human pose estimation on depth images. In (Criminisi et al., 2011a), regression forests are used for automatic localisation of anatomy in 3D CT images.

2.2 Privileged Information based Conditional Structured-Output Regression Forests

In this section, we present how we improve the standard regression forests by building better decision trees using privileged information and by learning shape constraints within the forest. We first learn higher quality decision trees using additional information. That additional information, like the head pose, that is only available at the training stage but not available at testing, is called privileged information. It can be used both in selecting the split function at some randomly chosen node and in learning the conditional voting model at the leaf node. We model the shape constraints between the locations of the different points within the forest. In contract to the traditional methods that learn one or several statistical shape models using global parametric representation, our method builds shape models at each leaf node. In this way, the shape models are naturally conditioned on the test images. The shape models can also be conditioned on the privileged information.

The learning stage is illustrated in Fig. 2.4. This includes the privileged information-based tree induction (2.2.1) and models-learning at leaf nodes (2.2.2). As shown, by randomly selecting a variable whose information gain is calculated, nodes decreasing the privileged information uncertainty and nodes decreasing displacement uncertainty, are interleaved in the decision tree. At each leaf node, three models are learned: First, a probabilistic model of the pdf of privileged information; Second, a regression model associated with each *base* feature point. A facial point is a base point for a certain leaf if the average relative offset of the patches that arrive at the leaf from the facial point in question is less than a threshold; Third, shape models related to the base feature point. Both of the latter two are conditioned on the privileged information.

During inference (described in sub-section 2.2.3), the privileged information is firstly estimated and then it is used in the subsequent steps for calculating the regression voting map and the structure constraint voting map, as shown in Fig 2.6. The final detection is carried out on the product of these two maps.

2.2.1 Privileged information-based tree induction

We pose the facial point localisation as a regression problem: given a set of input/output pairs (training data)

$$(x_1, y_1), ..., (x_M, y_M), x_m \in \mathcal{X}, y_m \in \mathcal{Y}, m \in 1, ...M,$$

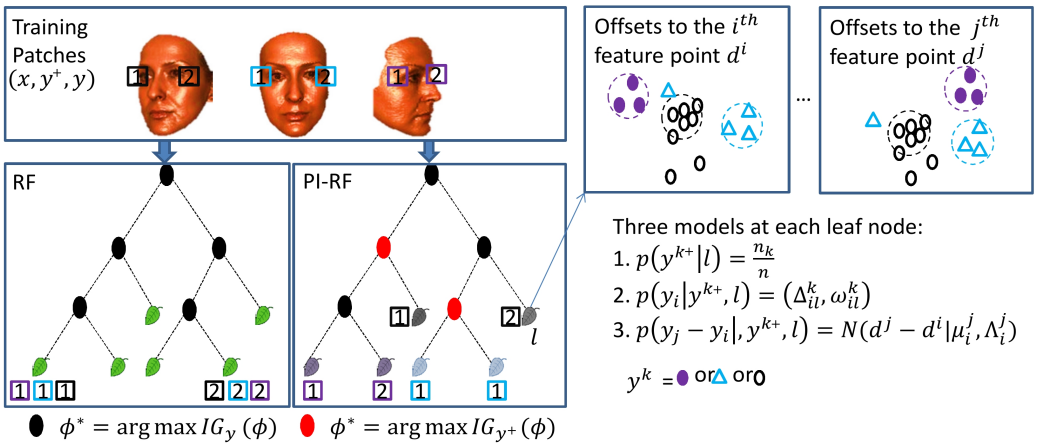


Fig. 2.4 An illustration of our proposed learning stage. The idealized tree induction for Privileged Information based Regression Forest (PI-RF) and Regression Forest (RF) is shown on the left. The training patches are from face images with a large variety w.r.t. the Privileged Information (PI) (here the head pose). A classical RF attempts to guide patches that are located around the same facial point to the same leaf node. However, as the example shows, the visual features vary due to changes in the PI and therefore it is difficult to guide them to the same leaf. On the contrary, in the PI-RF framework, the best split-function at some random internal nodes (in red) is selected directly according to the PI. As such, patches stored at the leaves tend to have low variation both in PI and in displacement. The information gain IG_y at dark nodes is calculated based on the entropy H_y, defined in Section 2.20 while at the color nodes, the information gain IG_{y^+} is calculated based on the entropy H_{y^+}, defined in Section 2.10. At each leaf node, one (or more) *base* feature point is defined and tree models are learned.

the goal is to find a mapping function $f: x \rightarrow y$ from a set of mapping functions $F: \mathcal{X} \rightarrow \mathcal{Y}$ with a small error on the prediction $\mathrm{y} = f(x)$. Similar to (Vapnik and Vashist, 2009), in our method, additional privileged information $\mathrm{y}^+ \in \mathcal{Y}^+$ is available during training as well. That is, the training set consists of triplets $(x, \mathrm{y}^+, \mathrm{y})$ instead of pairs (x, y). The privileged information $y^+ \in \mathcal{Y}^+$ belongs to a space that is different from the space $\mathcal{Y}$. The goal remains to find the best function $f: x \rightarrow y$ in the set of admissible functions.

In our case, a training sample is an image containing a face, the locations of facial points in the image and labels of privileged information, e.g., the head pose and subject's gender. Several fix-sized patches are randomly extracted from a training image, each represented by the image features $x = (x^1, x^2, ..., x^F) \in \mathcal{X}$ where F is the number of feature channels. Each patch is also annotated with a displacement vector $d = (d^1, ..., d^i, ...d^N) \in \mathcal{Y}$ to each of the N facial point and the privileged information label $\mathrm{y}^+ \in \mathcal{Y}^+$. The set of training patches is therefore given by $\mathcal{P} = \{\mathcal{P}_m = (x_m, d_m, \mathrm{y}_m^+)\}$. In this work each tree considers only one type of privileged information.

General tree growing procedure

A regression forest $\mathcal{T} = \{T_t\}$ is an ensemble of regression trees T_t. Each regression tree is most often induced greedily based on a randomly selected subset of the training data set $\mathcal{P} = \{\mathcal{P}_m\}$, in the following manner (Breiman, 2001). An empty tree starts with only one root node. Then, a number of test function candidates, ϕ, $\phi(x) \rightarrow \{0, 1\}$, defined over the image features x are sampled from a predefined distribution. Each patch is sent either to the left or to the right child depending on the test result. In this way, a test function ϕ partitions the training set into two sets, $\mathcal{P}_L(\phi)$ and $\mathcal{P}_R(\phi)$. Each candidate test function is evaluated according to a certain scoring function, e.g. *information gain*, so that high scores are assigned to splits that aid in predicting the output well, i.e. those that reduce the average uncertainty about the target. The best test function, that is the one with the highest score, is selected and stored at the node in question. Then, the training set is partitioned according to this test into two subsets that are propagated to the two children nodes. The same procedure is recursively applied at each child node. The procedure stops when certain criteria are met, typically, when there are fewer than a minimum number of examples or a maximum tree depth is reached.

Our binary test function $\phi_{f,R_1,R_2,\tau}(x)$ is defined as in (Gall et al., 2011):

$$\phi_{f,R_1,R_2,\tau}(x) = \begin{cases} 0 & \text{if} \quad x^f(R_1) < x^f(R_2) + \tau \\ 1 & \text{otherwise} \end{cases} \tag{2.5}$$

This is a comparison of the average value of the feature channel f in two asymmetric regions, R_1 and R_2, defined within the patch in question. $x^f(R)$ is the average value in region R and τ is a threshold.

Typically, the test functions are randomly generated and the one that maximizes the information gain $IG(\phi)$ that is achieved by splitting the data is selected. That is,

$$\phi^* = \arg\max_{\phi} IG(\phi) \tag{2.6}$$

The information gain is a popular criterion used to determine the quality of a split and has been used for both classification, regression and density estimation(Criminisi et al., 2011b). The information gain is the *mutual information* between the local node decision (left or right) and the predicted output and it is defined as follows,

$$IG(\phi) = \mathcal{H}(\mathcal{P}) - \sum_{s\in\{L,R\}} \omega_s \mathcal{H}(\mathcal{P}_s(\phi)), \tag{2.7}$$

where $\omega_s = \frac{|\mathcal{P}_s(\phi)|}{|\mathcal{P}|}$ is the ratio of the patches sent to the child node. $\mathcal{H}(\mathcal{P})$ is a measure of uncertainty on the set $\mathcal{P}$ and it is usually related to the entropy of the labels of the elements in the set. Depending on the nature of labels, $\mathcal{H}(\mathcal{P})$ can either be a discrete entropy or a differential entropy. We will address this in next section.

Entropy estimator

In our case, since $\mathcal{Y}$ and $\mathcal{Y}^+$ are different spaces, with different properties, an appropriate entropy estimator is needed.

For $\mathcal{Y}$, we use the class-affiliation method proposed by (Dantone et al., 2012b) to measure the uncertainty, that is defined as:

$$\mathcal{H}_{\mathcal{Y}}(\mathcal{P}) = -\sum_{i=1}^{N} \frac{\sum_m p(c_i|\mathcal{P}_m)}{|\mathcal{P}|} \log\left(\frac{\sum_m p(c_i|\mathcal{P}_m)}{|\mathcal{P}|}\right), \tag{2.8}$$

$$p(c_i|\mathcal{P}_m) \propto \exp\left(-\frac{|d_m^i|}{\lambda}\right), \tag{2.9}$$

where $p(c_i|\mathcal{P}_m)$ indicates the probability that the patch $\mathcal{P}_m$ is informative about the location of the facial landmark i. The class affiliation assignment is based on the Euclidean distance to the feature point. The constant λ is used to control the steepness of this function. In this way, we can avoid making a multivariate Normal distribution assumption on multiple feature points and calculate the differential entropy as in (Criminisi et al., 2011b).

So far as $\mathcal{Y}^+$ is concerned, we only consider discrete privileged information because: 1) for our problem it is difficult to obtain the ground truth of the continuous head pose for each face image; 2) learning the model conditioned on continuous variable is still not well studied (Sun et al., 2012). Therefore we discretise the head pose information by partitioning the pose space. In this context, head pose estimation becomes a multi-class classification problem. The finite set of privileged information classes is represented as $\mathcal{Y}^+ = \{1, 2, ..., K\}$. For each class, let h_k be the number of occurrences of the class, that is $h_k = \sum_{\mathcal{P}_m \in \mathcal{P}} \delta(y_i^+ = k)$. The empirical class probabilities $\hat{p}_k(\mathcal{P}) = \frac{h_k}{|\mathcal{P}|}$ (where $|\mathcal{P}| = \sum_k h_k$) are often used to calculate the entropy, i.e. $\mathcal{H}_N(\mathcal{P}) = -\sum_{k=1}^{K} \hat{p}_k(\mathcal{P}) \log \hat{p}_k(\mathcal{P})$ (see e.g. (Criminisi et al., 2011b) and references therein), however, it is pointed out by Nowozin (Nowozin, 2012) that the naive entropy estimator is biased and universally underestimates the true entropy. Therefore, as suggested in (Nowozin, 2012), we use the Grassberger entropy estimator (Grassberger, 2003), given as:

$$\mathcal{H}_{\mathcal{Y}^+}(\mathcal{P}) = \log |\mathcal{P}| - \frac{1}{|\mathcal{P}|} \sum_{k=1}^{K} h_k G(h_k), \tag{2.10}$$

where the function $G(h)$ is given by $G(h) = \psi(h) + \frac{1}{2}(-1)^h \left(\psi(\frac{h+1}{2}) - \psi(\frac{h}{2})\right)$, and ψ is the *digamma* function. For large h, the above function behaves like a logarithm and (2.10) is identical to naive entropy when $n \to \infty$. For small h, the estimation using (2.10) is shown to be more accurate.

In Eq. 2.20) and Eq. 2.10 we have designed the entropy estimator for both $\mathcal{Y}$ and $\mathcal{Y}^+$. During tree induction, at each internal node, the best split function is selected either based on Eq. 2.20) or Eq. 2.10. That is, the evaluation is either based on privileged information, or on the target. Note that in both cases the test itself is on the patch appearance, thus applicable both at training and test phase. When one of the stopping criteria of tree growing is met, several models will be learned at each leaf from patches that arrive there. An illustration of the tree induction process of our PI-based RF and of the traditional RF is in Fig. 2.4.

2.2.2 Models at leaf nodes

This section provides a description of our conditional regression model inspired by (Sun et al., 2012). More specifically, three models are learned at each leaf: 1) a probabilistic model of the pdf of the privileged information at the leaf; 2) a probabilistic regression model for the locations of the *base* facial points; 3) shape models that model the interdependencies of the locations of facial points that are neighbors of the *base* point in a predefined structure graph.

Probabilistic model of privileged information

First, at each leaf node, we calculate the pdf of the privileged information. Let n be the total number of training patches that arrive at a leaf node l, and let n_k be the number of patches belonging to class k. Then the probability for the class k at leaf l is

$$p(\mathrm{y}^{k+}|l) = \frac{n_k}{n}, \tag{2.11}$$

where y^{k+} is a shorthand notation that $\mathrm{y}^+ \in \mathcal{Y}^+$ belongs to the class k, i.e. $\mathrm{y}^+ = k$.

Conditioned regression model

Second, at each leaf node, we learn the conditional regression model for the *base* feature point. Our model shares tree structures for all states of privileged information. This is similar to the *Partial* conditional regression model proposed in (Sun et al., 2012). The samples are categorized into sub sets according to their privileged information labels and one conditional regression model is learned for each state.

Several regression models have been proposed in the literature. In our experiments we investigated two, both with one offset vector Δ and a weight ω, as the following.

1. A Mean Value model in which the offset vector Δ is the mean value of the offsets and the voting weight ω is defined as $\omega = |S_\Delta|^{-\frac{1}{2}}$ where S_Δ is the covariance matrix.

2. A Mean-Shift model in which the offset vector Δ is the mode of the largest cluster returned from a Mean-Shift algorithm applied on the corresponding set of patches that arrive at leaf node in question. The weight w is assigned as the relative size of the largest cluster.

This greatly reduces the model complexity and training time since we do not need to train and store separate random forest for each state of privileged information as in (Dantone et al., 2012b). Moreover, as shown in our experiments, it leads to better results.

The probability that the facial point i is located at y_i, given that a voting patch extracted at location z_x that arrive at leaf l is given by

$$p(\mathrm{y}_i|y^{k+}, l) \propto \omega_{il}^k \cdot \delta(||\Delta_{il}^k||_2 \leq \gamma) \tag{2.12}$$

where $\mathrm{y}_i = z_x + \Delta_{il}^k$, i and y^{k+} indicate the facial point number and privileged information state respectively. For notational clarity we will drop the facial point index i in the subsequent equations. γ is a threshold that prevents patches casting votes far away from place they

are extracted. This factor avoids a bias towards an average face configuration as the votes from long distant patches lack accuracy. Thus at each leaf, the regression-voting models are only valid for those patches whose mean offset is less than the threshold γ. In practice, each leaf is usually associated with one (in some cases two or more) facial point which we call a *base* point for the leaf in question.

Conditioned shape model

Third, at each leaf node, we learn the shape model for structured output regression. In contrast to the traditional face shape model such as ASM or CLM, our shape model is conditioned on the image information. Here we assume that the structure of the facial points can be organized in a graph, $G = (V,E)$, where V and E denote the sets of nodes and edges respectively. The nodes $i = 1,...,N \in V$ correspond to facial points and the edges $(i, j) \in E$ capture their spatial relations. The graph can either be dense or sparse or a tree structured model as (Zhu and Ramanan, 2012). In this work, we assume the graph structure is already known and what needs to be done is to parameterise it. In practice, we manually define a sparse graph model according to the physical proximity of the facial points.

Recall that each leaf is associated with one (or more) base points. We proceed to model shape constraints between the base point and its neighbours in the predefined structure graph. More specifically, assuming j is one of the neighbouring nodes of node i in graph G, i.e. $j \in Ne(i)$, their *relative* position is modelled as a Gaussian,

$$p(\mathrm{y}_j - \mathrm{y}_i | \mathrm{y}^{k+}, l) = \mathcal{N}(d^j - d^i | \mu_i^j, \Lambda_i^j) \tag{2.13}$$

Note that y^{k+} is the privileged information state and that the shape model is conditioned on it. One model is learned for each state. Recall that d^j and d^i denote the patch offset to the j-th and i-th point respectively. μ_i^j and Λ_i^j denote the mean value and covariance matrix of the Gaussian model.

2.2.3 Inference

During testing, patches from the test image are densely sampled from the whole image and sent down through all trees in the forest. A stride parameter is set to control the density of the sampling. Each patch is guided by the binary tests stored at the internal nodes and will arrive at one leaf of each tree in the forest. In what follows, we use I denote the test image data and let X be the set of image patches x extracted from the image. Let L denote the set of leaf nodes in the forest.

We now describe how to estimate the facial point locations and the privileged information state based on the models at leaves defined in section 2.2.2.

Privileged information inference

Similar to the ***MaxA*** approach in (Sun et al., 2012), the scoring function of privileged information state y^{k+} is defined as a sum of probabilistic votes contributed from all patches. Formally:

$$Score(\mathrm{y}^{k+}|I) = \sum_{x \in X} \sum_{l \in L} p(\mathrm{y}^{k+}|l) p(l|x) \tag{2.14}$$

where $p(l|x)$ is delta function that a patch arrives at a leaf node l (referred to as the leaf ID mapping probability). We then estimate the most likely state of the privileged information $\hat{\mathrm{y}}^{+}$ as:

$$\hat{\mathrm{y}}^{+} = \arg\max_{\mathrm{y}^{k+} \in \mathcal{Y}^{+}} Score(\mathrm{y}^{k+}|I). \tag{2.15}$$

This estimate will be used as a known variable in subsequent steps.

Independent regression

Firstly, we will describe the voting mechanism for independent estimation of locations of facial points, i.e. without considering the shape constraints. Similar to the *Partial Model* in (Sun et al., 2012), by expressing the probabilistic vote in terms of the distribution of each facial point for each *codeword* (leaf id) $p(\mathrm{y}_i|l)$ and the probability $p(l|x)$ that the image patch is mapped to a codeword, the scoring function conditioned on the privileged information is defined as

$$Score(\mathrm{y}|\mathrm{y}^{k+}, I) = \sum_{x \in X} \sum_{l \in L} p(\mathrm{y}_i|\mathrm{y}^{k+}, l) p(l|x) \tag{2.16}$$

Using the estimate $\hat{\mathrm{y}}^{+}$ of y^{+} given by (2.15), the best candidate of scoring functions over the privileged information state is selected as:

$$\hat{Score}(\mathrm{y}|I) = Score(\mathrm{y}|\hat{\mathrm{y}}^{+}, I). \tag{2.17}$$

Then mean-shift mode finding algorithm can be applied on the selected scoring function for the corresponding facial point.

Structured output regression

Second, we will describe how to infer structured output based on the conditional shape model in (2.2.2). Assume that a patch x that is extracted at z_x arrives at a leaf node l for

which i is one of the base points. The vote for the i-th point is cast at $\bar{\mathrm{y}}_i = z_x + \Delta_{il}$. Note that when privileged information is taken into account, Δ_{il}^k (instead of Δ_{il}) is used to estimate $\bar{\mathrm{y}}_i^k$ where k is the state of the privileged information given in (2.12). Here we drop the index k to simplify the notation and make this model more general for regular regression forests. Recall (see 2.2.2) that at each leaf we maintain shape models that model the relative locations of the neighbours $j \in Ne(i)$ for each base point. Then. given the estimate $\bar{\mathrm{y}}_i$ and the Gaussian model in (2.13), the structure constraint made on j is introduced in terms of the probability that the point j is located at y_j. The latter is modelled as $p_s(\mathrm{y}_j|\bar{\mathrm{y}}_i, l) = \mathcal{N}(\bar{\mathrm{y}}_i + \mu_i^j, \Lambda_i^j)$. Finally, the shape constraints on j given the estimated positions of all its neighbours i ($i \in Ne(j)$) are in the form of a scoring function $Score_s$ that gathers the votes cast by all the corresponding patches.

$$Score_s(\mathrm{y}_j|I) = \sum_{i \in Ne(j)} \sum_{x \in X} \sum_{l \in L} p_s(\mathrm{y}_j|\bar{\mathrm{y}}_i, l) p(l|x) \tag{2.18}$$

For each facial point, after accumulating votes cast from all patches in a test image, a local appearance evidence term like (2.17) and a structure constraint term like (2.18) are obtained. Then the structure constrained voting map is given as:

$$Score_v(\mathrm{y}|I) = Score(\mathrm{y}|I) \cdot Score_s(\mathrm{y}|I). \tag{2.19}$$

The mean-shift mode finding algorithm is applied on the final voting map to localize each facial point.

2.2.4 Evaluation

In this section, we present results on public datasets and compare with a number of methods in the literature. In comparison to the recent state-of-the-art methods, our method shows better or comparable result in terms of location accuracy and training efficiency.

We focus on datasets that contain face images that are recorded in uncontrolled environments, i.e. LFW. One representative dataset obtained at laboratory conditions BioID is also used for comparison.

Evaluation methodology

Throughout the experimental section, we measure the localisation performance using the inter-ocular distance (IOD)-normalized error. $e_i = \frac{||\mathrm{y}_i^D - \mathrm{y}_i^G||_2}{D_{IOD}}$. y_i^G is the ground truth location of point i, y_i^D is the estimated location of the point and D_{IOD} is the inter-ocular distance,

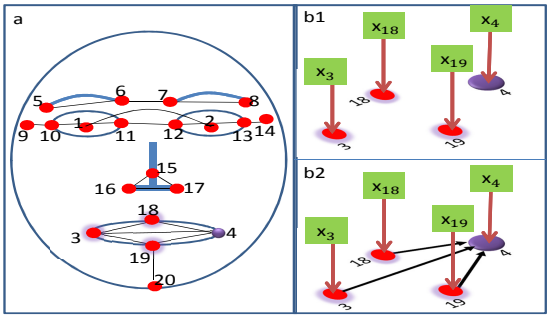


Fig. 2.5 Structured Output Regression. (a) shows manually defined sparse spatial relations of parts on face based on their physical locations. 20 selected face parts (dots) are displayed and their relations are represented by dark lines. The purple dot is one representative facial point and its neighbouring points are the red dots with purple shadow. (b1) illustrates an example of the independence assumption between points used in previous regression forests methods. Here we use x_i to represent the voting element x that is able to vote for part i, i.e. x arrive at leaves of which the i-th point is the base point. (b2) shows the spatial shape model of our method, in which the position of the 4^{th} point does not only depend on its voting patches x_4 but also on the estimated positions of its neighbouring points in the structure graph.

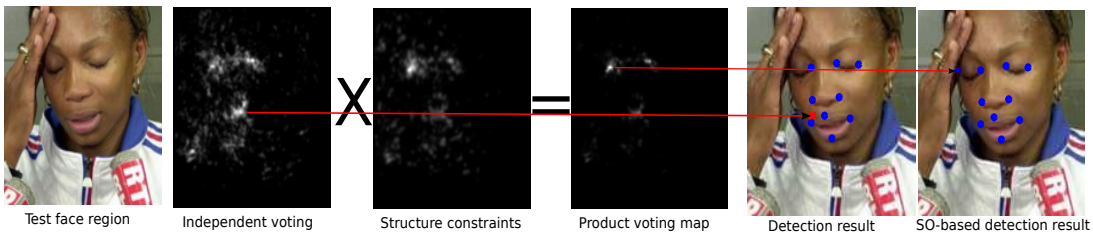


Fig. 2.6 An illustration of the structured output inference model. The face image shown here is *Laura_Flessel_0001.jpg* from LFW dataset.

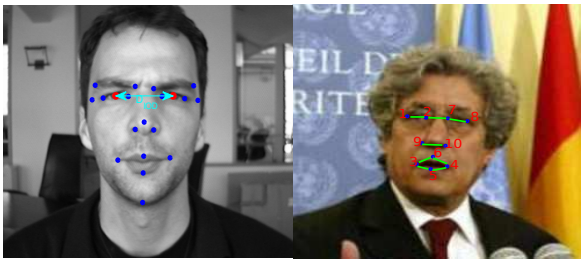


Fig. 2.7 Representative face images in BioID (left) and LFW (right) along with their facial point annotations. The green segments on the right face image represent our predefined graph model for the corresponding 10 facial points.

defined as the distance between the eye centres. Since the locations of the eye centers are not annotated in LFW dataset, the inter-ocular distance is calculated as the distance between the midpoints of the ground truth eye corners. A point is regarded as a correct detection if $e_i < 0.1$. This measure is used to calculate the successful detection rate in the experiments.

To evaluate the overall performance of localisation of multiple points on a face image, we use the m_{17} measure which defined in (Cristinacce and Cootes, 2008) as the mean error over all the internal points. Thus three of the 20 facial points, i.e. the chin and two temple points (i.e., P19, P9 and P14 in Fig. 2.5), are excluded when computing the m_{17}.

Experimental settings

As in most of the previous face points detection approaches (Cootes et al., 2012; Cristinacce and Cootes, 2008; Dantone et al., 2012b), our method assumes that the face bounding box is given both for training and testing images. The annotation of the LFW dataset already provides the face boxes for all face images. For the BioID dataset, we applied the Viola and Jones detector (Viola and Jones, 2004) in OpenCV to find the face bounding boxes (all bounding boxes are then resized to 125×125 pixels). The height is enlarged by 20% in order to ensure all facial point points are enclosed. In order to ensure a fair comparison, we keep the forest training setup in our experiments as similar as possible to the default setting of facial points detector described in (Dantone et al., 2012b). The key setting parameters include: maximum depth of each tree (20), test candidates at split node (2500), patch size ($0.25 \times$ face box size), image features (one channel of normalized gray values, 35 channels of Gabor features and 2 channels of Sobel features), number of patches per image sample (100). Unless stated otherwise, those parameters were used for forest training in all of our experiments.

In order to illustrate the benefits of using privileged information, we consider three types of privileged information, namely, yaw head pose, roll head pose and gender status for the LFW dataset. More specifically, we constructed the privileged information as follows: we use the discrete head pose labels for the yaw angle (left profile (20.3%), left (7.9%), frontal (42.4%), right (9.4%), right profile (20.0%)) provided by (Dantone et al., 2012b). Based on the locations of the facial points, we estimate the roll angles of head poses using the POSIT algorithm (DeMenthon and Davis, 1995) and discretise them into 3 labels (left tilt, upright, right tilt). We discard the pitch angle because it is difficult to get the ground truth for the face images in the wild. We also annotate the gender status (male, female) for each face image.

In order to evaluate the contributions of each component of our methods, we have built 24 forests using variations of the methods and tested on the LFW dataset (2.1). Below we

Table 2.1 Mean error of each facial point in LFW dataset (%).

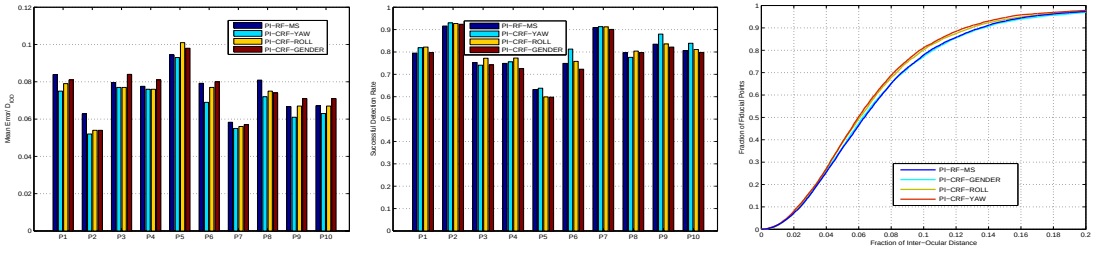
Forest ID	Short Description	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Avrg
F1	RF-MV	9.08	6.87	8.51	8.20	9.58	8.48	6.07	7.67	7.80	7.24	7.95
F2	RF-MS	9.29	6.72	8.23	7.85	9.40	8.23	5.65	7.84	6.95	6.89	7.70
F3	PI-RF-MV	8.35	6.65	8.15	7.78	9.37	8.22	5.93	7.51	7.51	7.12	7.66
F4	PI-RF-MS	8.39	6.28	7.96	7.76	9.44	7.92	5.83	8.09	6.67	6.71	7.51
F5	SORF-G	7.87	6.58	7.82	8.24	9.34	8.24	5.74	7.65	6.86	7.30	7.56
F6	SORF-MS	7.86	6.16	7.72	8.00	9.22	8.01	5.71	7.16	6.53	6.97	7.33
F7	PI-SORF-MV	7.72	6.45	7.61	7.93	9.04	7.96	5.67	7.41	6.77	7.20	7.37
F8	PI-SORF-MS	7.76	6.21	7.47	7.69	8.87	7.79	5.65	7.07	6.46	6.90	7.19
F9	CRF_YAW	7.70	5.30	7.90	7.90	9.40	7.10	5.60	7.50	6.20	6.40	7.10
F10	CRF_ROLL	8.60	5.40	7.80	7.80	10.10	7.60	5.60	7.70	6.70	6.70	7.40
F11	CRF_GENDER	8.10	5.40	8.50	8.10	9.70	8.20	5.70	7.30	7.20	7.00	7.52
F12	PI-CRF-YAW	7.50	**5.20**	7.70	7.60	9.30	6.90	5.50	7.20	6.10	6.30	6.93
F13	PI-CRF-ROLL	7.90	5.40	7.70	7.60	10.10	7.70	5.60	7.50	6.70	6.70	7.29
F14	PI-CRF-GENDER	8.10	5.40	8.40	8.10	9.80	8.00	5.70	7.40	7.10	7.10	7.51
F15	CSORF-YAW	7.00	5.30	7.30	7.60	8.20	6.60	5.30	7.10	6.00	6.50	6.69
F16	CSORF_ROLL	7.80	6.00	7.40	7.70	9.20	7.30	5.30	7.60	6.40	6.80	7.15
F17	CSORF-GENDER	7.80	6.20	8.10	8.30	9.10	7.90	6.00	7.70	6.70	7.30	7.51
F18	PI-CSORF-YAW	6.90	5.30	7.20	7.40	**8.00**	6.40	**5.00**	6.80	6.00	6.50	6.55
F19	PI-CSORF_ROLL	7.30	5.60	7.10	7.50	9.50	7.30	5.30	7.00	6.40	6.60	6.96
F20	PI-CSORF-GENDER	7.80	6.60	8.40	8.40	9.40	8.00	6.00	7.80	6.80	7.40	7.66
F21	PI-CSORF-Y+G	6.88	5.36	7.29	7.51	8.11	6.45	5.05	6.88	6.10	6.59	6.62
F22	PI-CSORF-R+G	6.91	5.47	7.18	7.29	8.87	7.42	5.27	7.00	6.35	6.67	6.84
F23	PI-CSORF-Y+R	**6.79**	5.22	**6.90**	**7.14**	8.10	**6.43**	5.19	**6.70**	**5.91**	**6.18**	**6.46**
F24	PI-CSORF-R+G+R	6.84	5.37	7.37	7.52	8.26	6.60	5.19	6.80	6.12	6.51	6.66
F25	*PI-CSORF-PIGT*	*6.70*	*5.30*	*6.70*	*7.14*	*7.76*	*6.32*	*5.00*	*6.60*	*5.71*	*6.11*	*6.33*

Table 2.2 Successful detection rate of each facial point in LFW dataset (%).

Forest ID	Short Description	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Avrg
F1	RF-MV	72.50	88.00	70.90	70.70	61.40	70.50	87.90	79.00	76.50	78.40	75.58
F2	RF-MS	77.20	88.40	72.30	73.00	63.70	72.90	90.80	79.60	82.00	80.70	78.06
F3	PI-RF-MV	77.20	90.00	73.20	73.00	62.50	71.70	89.20	80.70	78.90	80.00	77.64
F4	PI-RF-MS	79.50	91.60	75.30	74.90	63.20	74.90	90.90	79.60	83.50	80.60	79.40
F5	SORF-MV	79.60	90.40	74.60	70.50	62.50	73.00	91.60	77.50	82.60	78.50	78.08
F6	SORF-MS	81.50	91.00	75.80	73.20	65.40	73.30	91.70	79.10	85.30	80.80	79.71
F7	PI-SORF-MV	80.60	91.10	76.00	73.80	65.90	74.60	91.10	78.20	83.50	78.80	79.36
F8	PI-SORF-MS	81.50	91.30	75.30	74.90	63.20	74.90	90.90	79.60	83.50	80.80	79.59
F9	CRF_YAW	81.90	93.00	74.10	75.60	63.80	81.30	91.30	77.60	88.00	83.90	81.05
F10	CRF_ROLL	82.20	92.70	77.20	77.30	59.90	75.80	91.20	80.40	83.60	81.10	80.14
F11	CRF_GENDER	79.70	92.40	74.20	72.60	59.80	72.30	90.10	79.70	82.20	79.80	78.28
F12	PI-CRF-YAW	81.70	93.80	75.70	76.60	65.60	83.40	91.20	79.30	86.60	85.20	81.91
F13	PI-CRF-ROLL	81.70	92.80	77.50	76.80	62.10	76.70	91.00	80.00	83.60	81.70	80.39
F14	PI-CRF-GENDER	80.30	92.90	68.30	71.20	61.00	68.70	90.30	80.20	82.10	79.00	77.40
F15	CSORF-YAW	83.20	93.60	76.90	76.70	72.40	84.90	93.90	80.40	87.90	83.60	83.35
F16	CSORF_ROLL	80.90	92.30	78.80	75.60	65.80	79.30	93.50	77.90	84.40	80.90	80.94
F17	CSORF-GENDER	80.50	90.80	74.80	71.50	66.60	76.10	92.20	79.80	83.20	78.00	79.35
F18	PI-CSORF-YAW	83.40	**94.30**	78.80	77.20	74.10	**86.20**	94.30	81.70	87.50	83.90	84.14
F19	PI-CSORF_ROLL	83.60	92.90	80.70	78.70	65.20	78.50	94.30	82.70	84.90	80.90	82.24
F20	PI-CSORF-GENDER	79.80	91.40	75.10	71.50	65.20	74.90	91.80	79.80	82.90	78.80	79.12
F21	PI-CSORF-Y+G	83.20	93.60	77.70	75.90	**74.50**	85.00	**94.90**	81.80	87.70	83.30	83.76
F22	PI-CSORF-R+G	84.20	93.10	79.60	79.30	68.20	78.40	94.50	82.90	86.00	82.30	82.85
F23	PI-CSORF-Y+R	**85.10**	94.00	**82.20**	**79.40**	74.00	85.80	94.80	**83.50**	**88.40**	**85.80**	**85.30**
F24	PI-CSORF-R+G+R	84.80	92.80	79.40	76.20	72.40	84.50	94.40	83.00	87.80	83.80	83.91
F25	*PI-CSORF-PIGT*	*85.70*	*95.10*	*83.10*	*80.20*	*74.90*	*87.10*	*95.30*	*84.20*	*89.10*	*86.10*	*86.08*

describe the way in which the different variants are built . *RF-MV* creates the tree in a classical manner and at each leaf node, one single Mean Value model is learned. *RF-MS* also builds the tree in in a classical way but at each leaf node, a single Mean-Shift model instead of mean value model is stored. *PI-RF-MV* and *PI-RF-MS* are created using head pose yaw as privileged information and their leaf node models are the same as *RF-MV* and *RF-MS*. *SORF-MV* and *SORF-MS* are the structured output variants of *RF-MV* and *RF-MS* respectively. Their privileged information-based versions are *PI-SORF-MV* and *PI-SORF-MS* respectively. *CRF-YAW*, *CRF-ROLL* and *CRF-GENDER* are forests that conditional regression models are learned based on corresponding privileged information, head pose yaw angle, roll angle and gender status respectively while their *PI-* counterparts (i.e., *PI-CRF-YAW*, *PI-CRF-ROLL*, *PI-CRF-GENDER*) use privileged information during the tree building process. The following 6 forest, from F15 to F20 are the corresponding versions with additional shape models. All the above forests have the same number of trees (10). Each tree is trained using 1500 randomly sampled face images. The same random number generator is used for the same tree index of all the forests in order to make the comparison fair. Finally we construct four hybrid forests, from F21 to F24, that are used to evaluate the effect of fusing different types of privileged information (see Section 2.2.4). F25 shares the same forest from F24, however, during testing, it uses the ground truth privileged information to select the regression model at the leaf node.

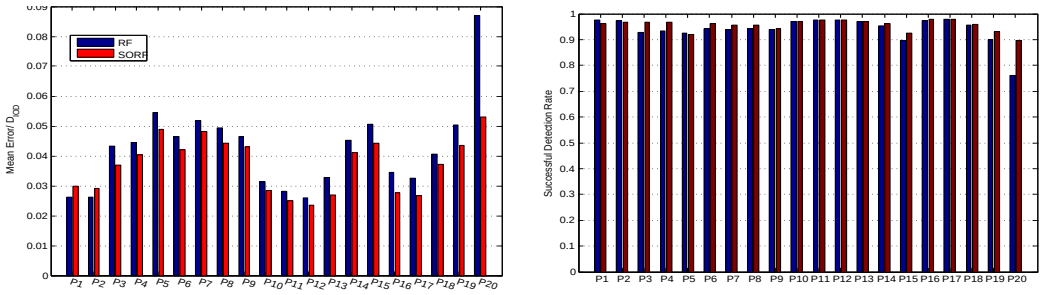
In the BioID dataset, we randomly select 400 face images for testing and the remaining 1121 images are used for training. Two different forests are built, each with 10 trees, one (SORF) with structured output while the other not (RF). Each tree is trained using 600 randomly selected images. The structure graph for 20 facial points in BioID dataset is shown in Fig. 2.5. For this dataset at each leaf node we use the mean shift-based voting scheme.



(a) Mean error (b) Successful detection rate (c) Cumulative Distribution

Fig. 2.8 Conditional model vs. single model. Some representative results on LFW dataset.

Experimental results



(a) Mean error (b) Successful detection rate

Fig. 2.9 Overall performance and comparison of RF and SORF on BioID dataset.

In what follows we summarize our results and discuss our findings from the experiments performed on the LFW and the BioID datasets. We evaluate the influence of the different components of our models and compare with the state-of-the-art methods.

Mean-Value vs. Mean-Shift As stated in 2.2.2, we have developed two voting schemes for the base point at each leaf, i.e. Mean-Value Model and Mean-Shift Model. We have conducted experiments on the LFW dataset in order to compare their performance in localizing of the facial points. By comparing the pairs: (F1, F2), (F3, F4), (F5, F6) and (F7 F8) in Table 2.1 and in Table 2.2, we conclude that Mean-Shift based voting scheme performs slightly better than Mean-Value model. On average, the difference is around 0.2% in terms of the mean localisation error and 1.96% in terms of the successful detection rate. In the remaining experiments we used Mean Shift-based voting.

Effect of Privileged Information In this part we will assess whether: 1) using the information gain on the privileged information as evaluation criterion at some internal nodes leads to better trained trees; 2) using regression model conditioned on the privileged information at leaf nodes is better. We assess the first by comparing forests trained using privileged information with their plain counterparts. We assess the second by comparing forests with conditional models at leaf nodes with their counterparts with single Mean Shift model at leaf node. In Table 2.3 we present results with and without using the head yaw as privileged information.

Furthermore, we assess the usefulness of three types of privileged information separately, i.e. head pose yaw angle, roll angle and gender status. As shown in Fig. 2.8, learning

Table 2.3 Comparison of Mean Error (ME) and Successful Detection Rate (SDR) of forests that using and not head pose yaw privileged information (%).

	Plain Training		PI-Training	
	ME	SDR	ME	SDR
Single Model	base line	base line	↓0.20	↑1.14
Conditional	↓0.62	↑3.31	↓0.76	↑4.14

Table 2.4 Estimation accuracy of privileged information.

Property	yaw (5 classes)	roll (3 classes)	gender (2 classes)
Accuracy	68.25%	85.10%	87.5%

models conditioned on head pose privileged information considerably outperforms the single model approach. Similar improvements can also be seen in Table 2.1 and Table 2.2 by comparing the mean error and detection accuracy of *F18*, *F19* with *F6*. The improvement in the mean error when using a conditional model is 0.78% and 0.52% respectively and the corresponding increase in the detection rate is 4.43% and 2.53% respectively. When using gender as privileged information, there is a 0.33% increase of the mean error and a 0.5% drop in the detection rate, however, for some facial points like P1 and P6, forests that use gender privileged information perform better. Further comparisons, as shown in Fig. 2.8 indicates that the gender privileged information does not have much impact on the model while the other two, i.e. head pose yaw and roll help to improve the performance.

Effect of Structured Output To evaluate the effectiveness of our proposed structured output (SO) method, experiments are conducted both on the BioID dataset and on the LFW dataset. For the experiments in the BioID dataset we used the structured graph with 20 nodes that is illustrated in Fig. 2.5 while for the LFW with 10 nodes is illustrated in Fig. 2.7 (right).

First, on the BioID dataset, we report the results from SO forests and non-SO forests, i.e. the comparison of the Regression Forest (RF) and Structured Output Regression Forest (SORF) in Fig. 2.9 in terms of the mean error and the detection rate. The comparison shows that our shape model reduces the mean error and increase the successful detection rate for most of the facial points. Particularly, the improvements of the difficult points like the chin point and lower lip centre are more significant. This is expected since these points are not located at intensity edges and therefore there is inherent uncertainty.

We perform several experiments on the LFW dataset, in order to compare SO-forests and non-SO-forests for several variants of our method. The results are shown in Table 2.1 and Table 2.2. We show the CDFs of the detection results for some representative forests in

Fig. 2.10. More details can be seen in the tables. The result validates the efficiency of our proposed structured output model in the localisation of the facial points.

Effect of Privileged Information Fusion Finally, we perform experiments in which we fuse different types of privileged information. *PI-CSORF-Y+G*, *PI-CSORF-R+G* and *PI-CSORF-Y+R* randomly take trees from two of the corresponding forests, i.e., *PI-CSORF-YAW* (Y), *PI-CSORF-ROLL* (R) and *PI-CSORF-GENDER* (G), 5 from each. *PI-CSORF-R+G+R* randomly takes 3 trees from each of the three corresponding forests. The CDFs of detection accuracy of the hybrid forests are shown in Fig. 2.11. Except the Y+R combination, the other fusion types have very similar performances, better than *PI-CSORF-GENDER* but not better than *PI-CSORF-YAW* or *PI-CSORF-ROLL*. This implies that the hybrid forests with trees trained based on gender privileged information do not lead to performance improvement. On the contrary, the hybrid forest, *PI-CSORF-Y+R*, with trees from YAW and ROLL forests outperforms both the YAW and ROLL forests.

Finally, we assess the prediction accuracy of the privileged information as shown in Table 2.4. We can achieve high accuracy in predicting the three types of privileged information. We also note that, F25 is able to achieve the most accurate result, if all the privileged information can be perfectly predicted.

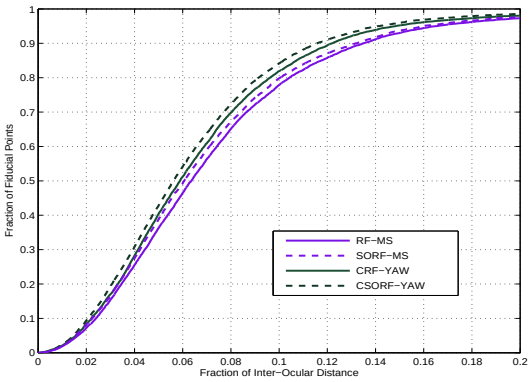


Fig. 2.10 Representative results from SO forests in LFW dataset, compared with their non-SO counterparts.

Run-time Performance We record the run-time performance on a standard 3.30GHz CPU machine. Our full method performs on LFW dataset at a average speed of 22 FPS while that of the baseline C-RF method is 25 FPS. Though we have more models at leaf nodes than C-RF, we estimate the privileged information within the forests, which is in contrast to C-RF that uses additional forests to estimate the conditional/privileged information.

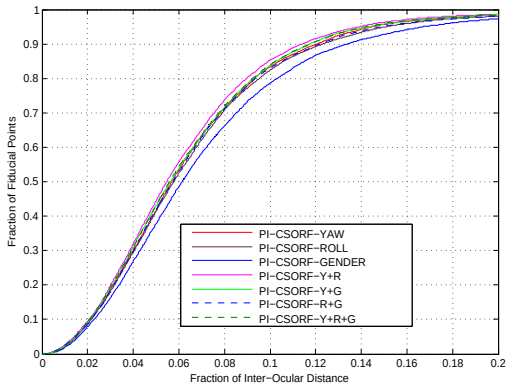


Fig. 2.11 The performances of hybrid forests on LFW dataset, compared with the original ones.

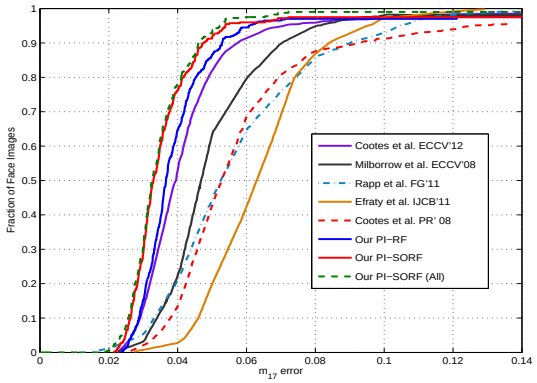


Fig. 2.12 CDFs of the m_{17} measure on BioID dataset, compared with reported results from (Cootes et al., 2012; Cristinacce and Cootes, 2008; Efraty et al., 2011; Milborrow and Nicolls, 2008; Rapp et al., 2011)

Comparison with state of the art

Finally, we compare our proposed methods with state-of-the-art approaches facial point localisation on the above mentioned datasets.

BioID Dataset On BioID, we initialize the detection using the OpenCV Viola and Jones face detector. Since related methods that start from the face bounding box have not discussed how they treat the failure cases of face detection (around 10 out of 400), we report the results by 1) manually defining the bounding boxes in the face images in which the face detection failed (the corresponding curves are with "All" label in the figures); 2) treating them as failure cases in facial point detection when calculating the successful detection rate and the cumulative distribution curve. In the literature, two types of curves are used to measure

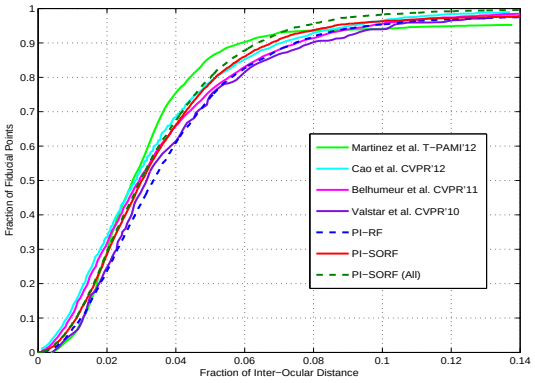
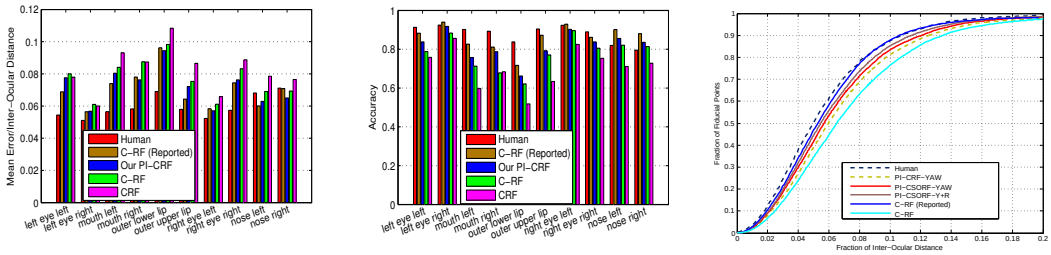


Fig. 2.13 CDFs over point error on BioID dataset, compared with (Belhumeur et al., 2011; Cao et al., 2012; Martinez et al., 2012; Valstar et al., 2010). For fairness, only 17 internal facial points are used.



(a) Mean error (b) Successful detection rate (c) Cumulative distribution

Fig. 2.14 Overall performance of our method on LFW dataset, compared with (Dantone et al., 2012b).

the overall performance. One is the commutative distribution function (CDF) over point error (i.e., fraction of points) and the other is the CDF of m_{17} (i.e., fraction of face images). They are shown in Fig. 2.12 and Fig. 2.13 respectively, together with results on the same dataset published elsewhere. As shown in these two figures, our method achieves very promising results on this dataset. Compared to the related method (Cootes et al., 2012) that has applied CLMs on the regression forest voting, our method performs better. This method has validated that its curve shape is consistent with the curve calculated from annotation with simulated Gaussian noise with around 1.5 pixels stand deviation. This implies that the root MSE of our method is smaller than 1.5 pixels. (Cootes et al., 2012) points out the distinctive "S" shape of our curve suggests that the errors in the localisation of different points are not correlated. The detection accuracy and the mean error for each of the 20 facial points is shown in Fig. 2.9.

LFW Dataset We now focus on the more challenging dataset LFW and compare with the regression forest method presented in (Dantone et al., 2012b). We use the publicly available implementation provided by the authors [1]. We have made a minor change, namely we changed the facial point data format from integer to float, in order to have a smoother error distribution. The CDFs of the error is shown in Fig. 2.14c. Note that the results that we obtained differ from what is reported in (Dantone et al., 2012b) possibly because the publicly available trained trees are a reimplementation. Different image features, parameter settings might affect the results. The close-to-human performance reported in (Dantone et al., 2012b) requires parameter optimization for each of the facial points and also training more than 10 trees in a sub-forest. The comparison here is based on the same experimental setting, namely the same number of training samples for each tree, the same image features used for training, and the same global parameters of a tree (maximum depth, number of testing candidates at each internal node). In this setting, our model outperforms the C-RF using the same yaw head pose privileged information. Furthermore, by incorporating the structure constraints and fusion of roll head pose information, the performance of our method is very close to human. As shown in Fig. 2.14a and Fig. 2.14b, the results are similar to results reported in (Dantone et al., 2012b) and very close to human performance.

We note that training our trees is computationally more efficient than training a C-RF. C-RF trains an additional forest for head pose estimation and also one forest for each head pose subset while only one forest is trained in our method. In the public implementation which we compare, 60 trees in total (10 trees for head pose estimation and 10 for each yaw pose) are built in C-RF while our method use only 10 trees in total. It also means that,

[1] http://www.dantone.me/projects-2/facial-feature-detection/

many more training samples are used in their model despite a tree is trained using the same number of training samples.

Cao *et al.* (Cao et al., 2012) have reported results on LFW87 (Liang et al., 2008) This is a dataset that is not publicly available but which seems to be of similar difficulty. We also list our MRSE (Mean Root Square Error) evaluation metric in Table 2.5 in order to give an idea about the relative performance but note that the results are on different datasets with similar characteristics.

Table 2.5 Percentages of test images with RMSE(Root Mean Square Error) less than the given thresholds on the LFW dataset, compared to (Cao et al., 2012; Liang et al., 2008) on LFW87 dataset.

RMSE	$<$ 5 pixels	$<$ 7.5 pixels	$<$ 10 pixels
Method in (Cao et al., 2012)	74.7%	93.5%	97.8%
Method in (Liang et al., 2008)	86.1%	95.2%	98.2%
PI-CSORF-Y+R	94.4%	96.3%	99.2%

LFPW Dataset We compare our method and the C-RF detector on test images from the LFPW dataset to test whether the learned models can be transferred to a different dataset. Again, the OpenCV Viola and Jones face detector is applied first. The mean error of each facial point is shown in Fig. 2.15. Although our detector does not perform as well as (Belhumeur et al., 2011) and (Cao et al., 2012), the average mean error, around 2 pixels, is very low. It is worth noting that neither our model nor C-RF is trained on LFPW and it is known that the image quality of LFW is much worse than that of LFPW. The performance of our detector and C-RF on LFPW is close to their performance on LFW. When the error fraction is less than 0.1, a detection is regarded as success. We reported the successful detection rate of each facial point in Fig. 2.16. As it can be seen, for most of the points, the successful detection rate is very high, more than 90%. The mouth corners and the outer lower lip are the most difficult points to localize. In Fig. 2.17, we show the detection results of our model and of the C-RF detector on some example images from LFPW . As it can be seen, under partial occlusion, both C-RF and our CRF method fail to localize all points at the correct positions since they are both local detectors. On the contrary, CSORF method is able to handle such cases since it takes the structure constraints into account.

AFLW Dataset Finally we show the performance on the AFLW dataset and compare to recent Regression Forests based methods including the baseline C-RF (Dantone et al., 2012b), RF-CLM (Cootes et al., 2012) (RF combined with CLM) as shown in Fig. 2.18. We

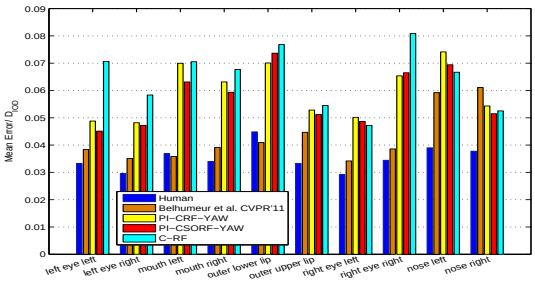


Fig. 2.15 Mean error of our model on LFPW dataset, compared to C-RF detector from (Dantone et al., 2012b).

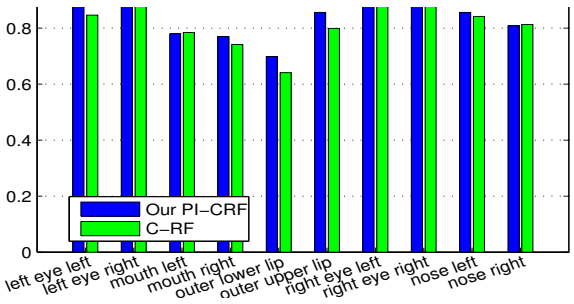


Fig. 2.16 Successful detection rate of our model on LFPW dataset, compared to C-RF detector from (Dantone et al., 2012b).

Fig. 2.17 Example Images from LFPW dataset. First column shows detected facial points by C-RF (Dantone et al., 2012b), second column the detection results by PI-CRF-YAW forest and the last column the detected facial points by PI-CSORF-YAW Forest.

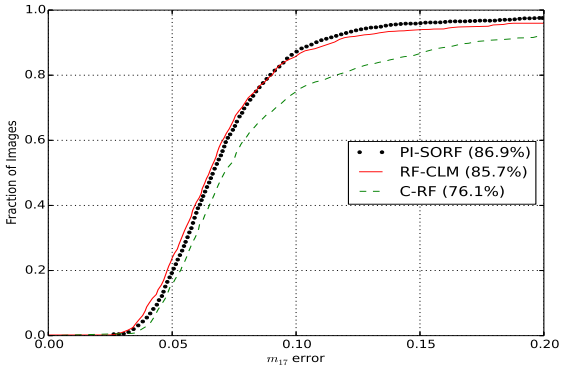


Fig. 2.18 Comparison to RF based methods (C-RF (Dantone et al., 2012b) and RF-CLM (Cootes et al., 2012)) on AFLW.

select 1000 images from AFLW for testing and the rest of them for training the forests and repeat this process for four times and we report the average results as shown in Fig. 2.18. Our proposed method performs significantly better than the baseline RF and on par with the RF-CLM, which has explicit shape models. Our method is able to further combine with other shape models for performance boost.

Sensitivity to Face Bounding Box Shift In recent years, the cascaded methods have shown promising results in facial points detection. However, compared to the local-based methods as ours, they are more sensitive to initialization, which is often calculated from face detection. It is because the features are extracted around the initial estimate of the landmarks. Applying a different face detector influences the results of the cascaded methods - this is evident by the fact that (Burgos-Artizzu et al., 2013; Cao et al., 2012) rely on multiple initializations or to the so called 'smart starts'. By contrast, the method is this work is a local-based one that does not rely on any initialization shape: patches within the bounding box will be used and the RF will decide which ones will vote for which landmark. This decision is based on the patches appearance and not on their distance from a shape. Indeed, regions near the facial points give better predictions however such information (i.e. the true distance) is not known at test time. When the bounding box shifts due to an inaccurate face detection, then some patches fall out of the new bounding box, and some new patches fall in. However, all the patches that are in the intersection of the old and the new bounding box will vote in exactly the same way. This makes local methods more robust.

We perform experiments on LFW to demonstrate this. We apply the state of the art cascaded method, SDM (Xiong and De la Torre, 2013) on the same test images from LFW that we have used and report the results of their common facial points. We shift the face

Bounding Box (BB) by 5% to 20% and the results are as follows: Though SDM and our

Bounding Box shift	0%	5%	8%	10%	20%
SDM Mean Error	6.45	7.71	15.56	22.57	40.36
Our Mean Error	6.46	6.48	6.51	6.70	9.20

Table 2.6 SDM (Xiong and De la Torre, 2013) vs. our method when face BB shifts.

method have similar results given the ground truth face bounding box, when face bounding box shifts, the performance of SDM drops rapidly. On the contrary, until the shift is very huge (20%) and results in some facial points obviously fall out of the face bounding box, our method is fairly robust to the bounding box shifts.

2.3 Fine-tuning Regression Forests Votes

In previous section, we have focused on the training stage and presented a scheme that constructs better regression forest by using privileged information and by learning pairwise constraints within the forests. However, as the regression forest works in a sliding window fashion, one local patch will be transformed to one vote in a tree. The number of votes from one test image is usually very large and not all of them contain valid voting information. Therefore, in this section, we focus on the testing stage by analysing the votes from regression forests. We present a regression forests votes fine-tuning scheme, by sieving and/or aggregating. In order to filter out invalid votes, we pass them through several sieves, each associated with a discrete or continuous latent variable. The sieves filter out votes that are not consistent with the latent variable in question, something that implicitly enforces global constraints. In order to aggregate the votes when necessary, we adjusts on-the-fly a proximity threshold by applying a classifier on middle-level features extracted from voting maps for the object landmark in question. Moreover, our method is able to predict the unreliability of an individual object landmark. This information can be useful for subsequent object analysis like object recognition. This framework is tested on two object alignment tasks, face alignment and car alignment, on datasets with challenging images collected *in the wild*.

2.3.1 RF votes with latent variable

For tree construction, we use the procedure as we used in previous section 2.2.1. We briefly recall it as follows. In order to calculate the information gain (IG) for split function selection, an entropy-like class uncertainty $\mathcal{H}(\mathcal{I})$ on a set of image patches $\mathcal{I}$ is used, which is defined as:

$$\mathcal{H}(\mathcal{I}) = -\sum_{\mathrm{y}_j \in H} \frac{\sum_{I_i \in \mathcal{I}} p(\mathrm{y}_j|I_i)}{|\mathcal{I}|} \log\left(\frac{\sum_{I_i \in \mathcal{I}} p(\mathrm{y}_j|I_i)}{|\mathcal{I}|}\right), \tag{2.20}$$

$$p(\mathrm{y}_j|I_i) \propto f(|d_{\mathrm{y}_j}^{I_i}|) = \exp\left(-\frac{|d_{\mathrm{y}_j}^{I_i}|}{\alpha}\right), \tag{2.21}$$

where $p(\mathrm{y}_j|I_i)$ indicates the probability that the patch I_i belongs to the the j-th landmark (Dantone et al., 2012b), $j \in 1,...,J$. We use y_j to denote the location of the landmark. $f(\cdot)$ is a function that transforms the Euclidean distance $d_{\mathrm{y}_j}^{I_i}$ into a proximity metric. This proximity metric is used throughout this section. The constant α controls the steepness of this function. Note that the distance measure d is normalized by the object size.

Once the regression forest is trained, the observations, i.e. image patches $I_i \in \mathcal{I}$ are extracted from the testing image location y_i and fed to it. When they arrive at leaf nodes they

cast weighted votes $v(h|I_i)$ for the location of one or more landmarks. For a given hypothesis $h \in H$, the score of h is determined by the sum of votes that support the hypothesis: $Score(h) = \sum_i v(h|I_i)$. In practice each patch I_i will be sent to each tree $t \in T$ in the forest, i.e. $Score(h) = \sum_i \sum_t v(h|I_{it})$. We will drop the t in the subsequent discussion for clarity and consistency with other methods.

With the procedure described above, there are some votes that are inconsistent with some latent variables, for instance, in head pose and face center. These votes are unlikely to vote correctly. Some previous work (Razavi et al., 2012) proposes to augment the hypothesis space by a latent space Z to enforce consistency of the votes in some latent properties $z \in Z$. That method can only deal with discrete latent variables and has high memory requirements and computational complexity, when large training data is used since all training patches need to be stored. By contrast, the latent space in our work can be either discrete or continuous. The score of a hypothesis in the augmented space is then given by:

$$Score(h) = \sum_{i, z \in \phi(\hat{z})} v(h, z|I_i) \tag{2.22}$$

where $\phi(\hat{z})$ is an affiliation term defined as follows. When the latent space is discrete, $\phi(\hat{z}) = \{\hat{z}\}$, with $\hat{z}$ a discrete label. It means votes with the same latent variable as $\hat{z}$ are used. When the latent space is continuous, $\phi(\hat{z}) = \{z : ||z - \hat{z}|| \leq r\}$, where r is the radius of a region around $\hat{z}$. The details are described in Section 2.3.2.

2.3.2 RF votes sieving

Sieving via discrete latent variable

Our method of sieving votes using a discrete latent variable is similar to the conditional regression forest **Partial Model** proposed in (Sun et al., 2012) that was used for human pose estimation. During training, each patch extracted from the training samples is annotated with a discrete latent label. We use the tree construction procedure proposed in 2.3.1. When the training patches arrive at the leaf node l, we learn one model for each state of the latent variable. More specifically, we first partition the training patches according to their latent variable labels and then learn a model in each partition with latent label z for the hypothesis h. The model vector is $(\Delta_l^z, \omega_l^z, p_l^z)$. where Δ_l^z is the relative offset vector, obtained by taking the center of the largest mode found by mean-shift clustering method (Comaniciu and Meer, 2002) in the partition with latent label z, similar to (Sun et al., 2012). ω_l^z is weight, given by the relative size of the largest cluster. For the latent variable z, p_l^z is the probability of the latent variable at leaf node l, that is calculated as the proportion of the training samples

whose label is z, that is,

$$p_l^z = \frac{n_l^z}{\sum_{z \in Z} n_l^z} \tag{2.23}$$

where n_l^z is the number of training patches with the latent label z. When a patch I_i extracted from the location y_i arrives this leaf node l, the vote is represented as:

$$v(h,z|l) = \omega_l^z \delta(\Delta_l^z + y_i - h). \tag{2.24}$$

Since the probability of the latent variable is independent of the hypothesis, its scoring function is:

$$Score(z) = \sum_i v(z|I_i) = \sum_i \sum_{h \in H} v(h,z|I_i) = \sum_l p_l^z. \tag{2.25}$$

The latent variable is estimated as $\hat{z} = \arg\max_{z \in Z} Score(z)$. Given the estimation, the hypothesis scoring function is formed by using the votes with the corresponding latent state, i.e.,

$$Score(h) = \sum_{i, z \in \phi(\hat{z})} v(h,z|I_i) \tag{2.26}$$

Sieving via continuous latent variable

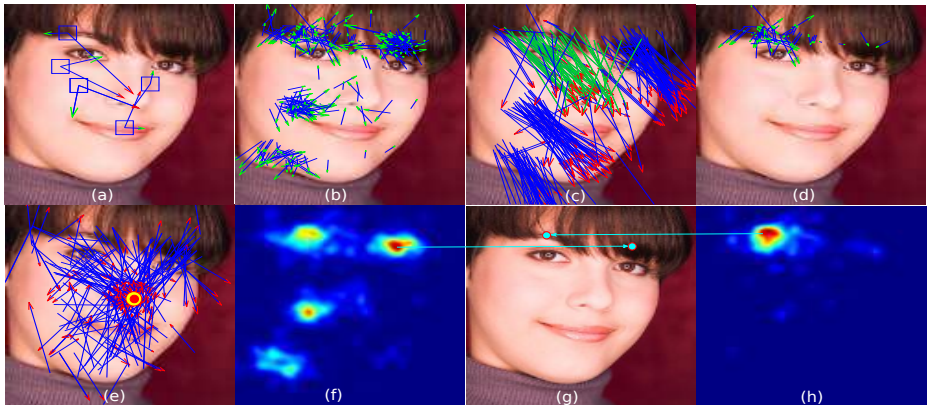


Fig. 2.19 Illustration of sieving via continuous latent variable (face center). (a) A voting element consists of two offset vectors, one to the target point (green arrow) and the other to face center (red arrow). (b) Original set of votes for the left brow center. (c) The absolute face center votes, those in green are regarded as consistent to the face center. (d) The remaining voting elements filtered by the face center sieve. (e) All voting elements are used to localize the face center (red dot). (f) and (h) are the Hough maps generated from votes of (b) and (d) respectively. (g) shows the corresponding detection results.

The tree construction process of the sieving via continuous latent variable is similar. Each training patch is associated with a continuous latent variable label, for instance the

displacement to the object center. This latent information is not used until the patches arrive the leaf node. We use the face center as an example to show how a continuous latent variable is modelled. The leaf model vector is $(\Delta_l^z, \omega_l^z, \Delta_l^{cz}, \omega_l^{cz})$. In addition to Δ_l^z and ω_l^z, we have two similar terms, Δ_l^{cz} and ω_l^{cz}, that are the offsets to the face center and the corresponding weight respectively, learned in a similar way of learning Δ_l^z and ω_l^z.

During testing, we will first estimate the state of the latent variable z, i.e. the location of the face center. Similar to calculating the actual voting of the hypothesis, the absolute voting to the face center in return is $y_i + \Delta_l^{zc}$, which is the actual form that is accumulated into the Hough space. The voting function is calculated like in Eq. 2.24. Thus the score function is:

$$Score(z) = \sum_i v(z|I_i) = \sum_i \sum_{h \in H} v(h, z|I_i) \tag{2.27}$$

Then a mean-shift (Comaniciu and Meer, 2002) algorithm is employed on the Hough map to find the mode. This is used as an estimate of the latent variable, $\hat{z}$. We then define a region around $\hat{z}$ as

$$\phi(\hat{z}) = \{z : ||z - \hat{z}|| \leq r\} \tag{2.28}$$

The radius r is learned at training time. The sieve filters out the patches which cast votes out of this region, i.e., retains only the votes that are consistent with the estimate of the latent variable. The voting model for the hypothesis is the same as described in Eq. 2.24. The score function of hypothesis h after the latent continuous sieve can be written as:

$$Score(h) = \sum_{i, z \in \phi(\hat{z})} v(h, z|I_i) \tag{2.29}$$

It shares the same form of Eq. 2.26 but with different $\phi(\hat{z})$ property since z here is in continuous space.

As shown in Fig. 2.19d, after filtering by the sieve, voting elements that violate the face center consistency and vote for other face center hypotheses, are removed from the votes set. The ones that satisfy the face center consistency are kept.

2.3.3 RF votes aggregating

Taking all the voting elements into account for each hypothesis can lead to bias towards the mean shape and also it is very time consuming. Thus in practice, when collecting the votes for an individual feature point, a threshold is applied, similar to (Dantone et al., 2012b; Kontschieder et al., 2012; Sun et al., 2012). This works as a filter that prohibits votes with large offsets. This threshold is typically optimized during training and kept fixed during

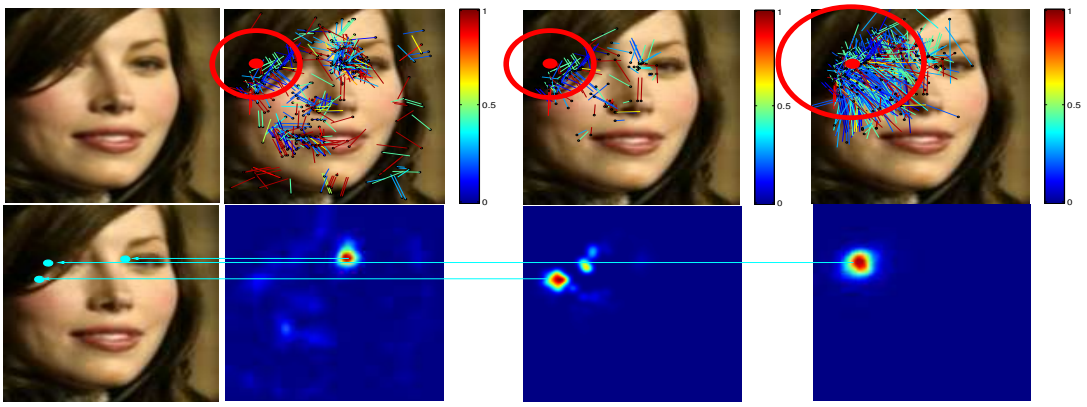


Fig. 2.20 Illustration of aggregating the votes by updating the threshold. From left to right, the first row shows the original face image, all votes for the point ($\lambda = 0.35$), votes passed face center sieve and the aggregated votes from updated threshold ($\lambda = 0.22$) passed face center sieve. The color represents the weight of each vote and the dark terminal is the voting destination. The second row shows the detection results, normalized Hough map for original voting, after face center sieving and re-voting.

testing. Only the votes that satisfy a threshold are allowed to vote for the hypothesis, i.e., $f(\Delta) > \lambda$, where $f(\cdot)$ is the proximity function defined in Eq. (2.21).

This mechanism works well in most cases but fails, for example when a feature point is heavily occluded. As shown in Fig. 2.20, in the presence of a heavy occlusion, only few valid voting elements remain after the face center sieve is applied. This is expected since in the case of heavy occlusion, there are no patches near the occluded facial landmark that can cast reliable votes. In such cases, we should allow votes from patches that are farther away. Thereby we need to reduce the proximity threshold. Such votes, introduce implicit facial shape constrains. In order to determine an image-dependent proximity threshold λ_j for the j-th landmark, we pose it as an rare event detection problem using one class SVM (OC-SVM) (Chen et al., 2001), which, given a certain value λ_j and the voting map calculated using that threshold, determines whether the threshold should be decreased or not. In order to train an OC-SVM for each facial point, we collect a set of positive training instances, i.e., the images in which the facial point is not occluded and can be localized accurately using the current proximity threshold. We propose to use middle-level features that are extracted directly from the votes set after the object center sieve is applied. The feature is represented as a histogram of the voting orientation. Specifically, we first compute the voting center using a mean-shift algorithm, then the votes are divided into four separated sub-windows using the x-y coordinate system originated at the voting center. Then we calculate the voting orientation histogram in each sub-window, with 12 equally divided bins, i.e., 30° per bin. This results in a 48-dimensional feature denoted by x_1. As shown in Fig. 2.21, the histogram

of voting orientation of occluded facial points (the right one) differ significantly from non-occluded ones (the left two). By contrast, the histograms of non-occluded landmarks are similar, despite the fact that the face images are quite different. Given the features of positive training instances for each facial point, a RBF-kernel based OC-SVM model is learned that is able to determine whether or not to adjust the proximity threshold.

We also calculate another feature, that is the ratio of votes after and before the face center sieve is applied, $x_2 = \frac{|V^F|}{|V|}$. V and V^F are respectively the set of votes before and after the face center sieve is applied. If x_2 is less than a threshold τ, then the proximity threshold should be reduced. In order to determine how much the proximity threshold should be reduced for a certain facial landmarks, we consider whether our classification scheme has determined that the threshold for neighbouring landmarks should be reduced or not as well. This is an indicator that the corresponding patches around them are also unreliable (e.g., there is occlusion). Therefore the proximity threshold reduction should be larger. We define two neighbours $j' \in Ne(j)$ for each landmark. The votes aggregating, or proximity threshold updating procedure is summarized in Algorithm 1.

Algorithm 1 Aggregating votes

Input: $\Lambda = \{\lambda_j\}$ with pre-optimized proximity thresholds
Output: Updated proximity thresholds Λ

initialize the update index vector $K = \{k_1, ..., k_j, .., k_J\}$ with all zeros ▷ # of steps to update
for all $j \in \{1, ..., J\}$ **do**
    collect voting elements V_j based on λ_j
    apply face center sieve and obtain V_j^F
    calculate the middle level feature x_1 and x_2
    $Rt \leftarrow \text{svm}_j(x_1)$ ▷ apply the OC-SVM
    if $Rt == -1$ **or** $x_2 < \tau$ **then** ▷ τ,threshold
        $k_j := k_j + 1$
    end if
end for
for all $j \in \{1, ..., J\}$ **do**
    for all $j' \in Ne(j)$ **do**
        if $k_{j'} > 0$ **then**
            $k_j := k_j + 1$
        end if
    end for
    $\lambda_j := \lambda_j - k_j * step * \lambda_j$ ▷ $step$=0.3
end for

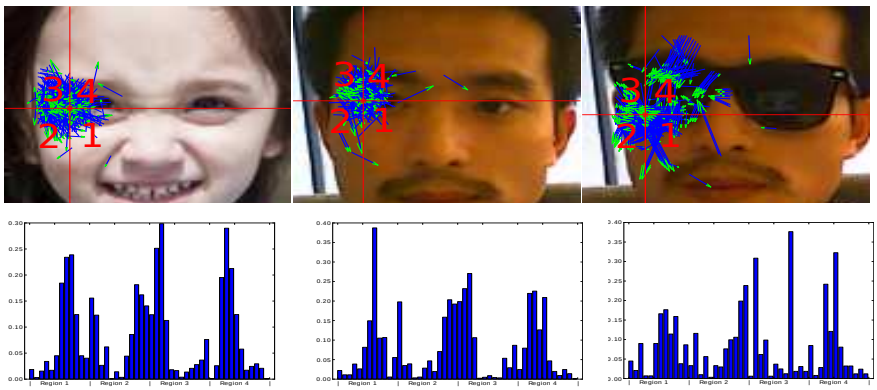


Fig. 2.21 Feature extracted from the votes passed the face center sieve. The left shows an example image for training the one class SVM classifier for the left eye corder. The middle shows an example tested positive and the right shows an example tested as outlier. The red lines split the votes into four regions and the below shows their corresponding features, i.e., x_1.

2.3.4 Landmark unreliability

Face analysis systems, for instance face recognition and facial expression, suffer a lot from the partial occlusion caused by hair, hand, sunglasses, scarf or other objects. Ekenel and Stiefelhagen (Ekenel and Stiefelhagen, 2009) has studied the face recognition performance drop due to partial occlusion. In general, occlusions can lead to two main problems. First it leads errors in the object alignment and second it leads to the extraction of features from places that do not contain facial information. Most of alignment methods give out the result as is. The subsequent feature extraction step can only assume 100% correctness of the alignment and equal importance of the features from different landmarks. In our method, we estimate the unreliability of each landmark, as discrete level, ranging from 0 to 3, 4 levels in total, which shares the value of threshold updating index k_j calculated from Algorithm in 4. A larger value means higher unreliability and landmarks with $k_j = 0$ are reliable. Note our method compensates for the cases where nearby votes are not reliable (i.e., $k_j > 0$). $k_j > 0$ does not necessarily mean occlusion is presented. We argue that even though the points under heavy occlusion can often be localized in a high accuracy, the features extracted from the nearby region are not reliable for further object analysis, such as face recognition or car reconstruction.

2.3.5 Experiments

To evaluate the efficacy of the proposed method, we conduct experiments on two face databases (LFW, AFLW) and one car database (CMU-CW), both collected in uncontrolled

environments. Object alignment in both cases is challenging since a) most car/face landmarks are only weakly discriminative for detection; b) the images are taken from various viewpoints and c) often contain cluttered backgrounds and severe partial occlusion.

Implementation details

Forest Model on the LFW We use the trained model from (Dantone et al., 2012b), denoted by **CRF** in this work, on the LFW as a baseline for comparison. At each leaf node, the trained model provides offset vectors to 10 facial points and it also provides a mean patch offset vector to the center of the bounding box. The latter is treated as a continuous latent variable for sieving the votes, i.e. Δ_l^{cz} in our work. We assign a unit weight to each vote, i.e. we set $\omega_l^{cz} = 1$. We denote this forest model with the bounding box center as continuous latent variable by **CRF_C**. This allows us to evaluate our contributions using the CRF as a baseline.

Forest Model on the AFLW We show the contribution of each component of our method on AFLW by training models that are listed in Table 2.7. The trees in the forests F1, as in (Dantone et al., 2012b) are trained without using any additional information. In order to train forests with sieves using latent discrete variable, i.e. F2-F5, we quantize the head yaw angles of the training samples into 3 labels like (Dantone et al., 2012b). We train a forest using the additional discrete information to learn multiple voting models at leaf nodes as described in Section 2.3.2. A similar idea is proposed by Sun *et al.* (Sun et al., 2012) for human pose estimation. We denote their method by CRF-S. In forests F3, F4 and F5, each vote at leaf node contains voting information to the face center as described in Section 2.3.2. In the forest model of F4, we set the proximity threshold of an individual facial point to 0, i.e. allow all the votes from the face to vote for the facial point. The tree model of F5 is the same as F4 but performs threshold adjustment as described in Section 2.3.3. We use

Table 2.7 Description of forest models trained on the AFLW

Forest ID	Sieves		Aggregation
	Discrete	Continuous	
F1	No	No	No
F2 (CRF-S)	Yes	No	No
F3	Yes	Yes	No
F4	Yes	Yes	Max. aggregating
F5	Yes	Yes	Yes

the same macro settings of the forests of (Dantone et al., 2012b) such as the image features, maximum tree depth (20), number of tests at the internal nodes (2500), forest size (10 trees

in total) and the bandwidth of the mean-shift algorithm. Also we use the same random subset of the training samples for the same index of tree in each forest in order to avoid a bias caused by random sampling of the training data.

Throughout our experiments we report the root mean square error (RMSE) of the localisation of the landmarks with respect to the manually labelled ground truth landmark locations. The error in the face images is normalized as a fraction of the inter-ocular distance as in (Cao et al., 2012; Dantone et al., 2012b; Martinez et al., 2012).

Forest Model on the CMU-CW We train one forest for each of the 5 views using the training set-up used in (Boddeti et al., 2013). We randomly select 400 images for each view for training and use the rest for testing. We sample 30 patches sized 30×30 from a non-occluded landmark region for training. We use the car center, calculated as the mean value of all the landmarks, as a continuous latent variable in this model. A tree in the forest is trained on 300 randomly sampled car images and 4 trees in total are trained for each view.

Parameters for votes sieving The key parameter associated with the continuous variable sieve, the radius r, is set to 0.3 through a grid search on the AFLW validation set. We use the same sieving parameters for LFW and CMU-CW.

Parameters for votes aggregating For each facial landmark, we select the most accurate 500 detections (localisation error less than 0.1) from the AFLW validation dataset as positive training samples to train the OC-SVM model. When there are not enough training samples we select some from the training samples. The OC-SVM models of the CMU-CW is directly trained on the training samples. We use the LibSVM (Chang and Lin, 2011) to train the OC-SVM model.

Method evaluation

In this section we evaluate the influence of the different components of our models and summarize our findings from the experiments performed on the AFLW dataset. We repeat the experiment for 4 times. The reported results below are averages over the 4 runs.

Performance of votes sieving Since the sieving can be based on both discrete and continuous latent variables, we evaluate them separately.

Sieving via a discrete latent variable. In order to evaluate the efficacy of sieving via a discrete latent variable, i.e. the discrete head pose in our case, we report the results using forests F1 and F2. As can be seen from Fig. 2.22 where the cumulative distribution of facial points over error threshold is shown, the forest with sieves associated with the discrete head pose label performs significantly better. However, neither of them is able to deal with challenges like occlusion and shadows, and only a proportion of the facial points can be

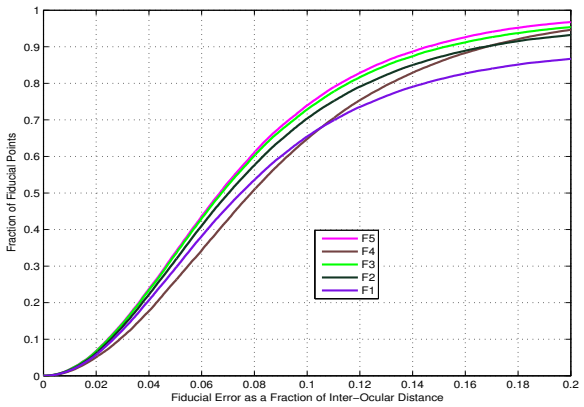


Fig. 2.22 Error distribution.

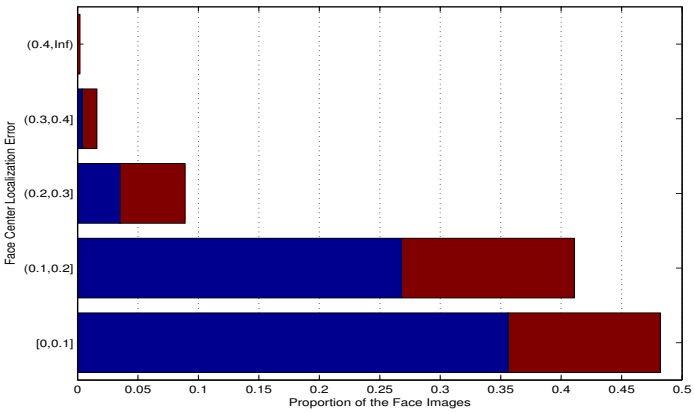


Fig. 2.23 Face center estimation error distribution.

localized with high accuracy. The percentages of facial points with error less than 0.1 are respectively 65% (F1) and 70% (F2).

Sieving via continuous latent variable. Since sieving using continuous latent variables involves estimating the latent variable, we first evaluate the stability of the estimation by measuring the error, in our case, the face center localisation error. As is shown in Fig. 2.23, though the localisation is not highly accurate, the performance is very stable: only 2 out of the 1000 test images have localisation error larger than 0.4 and more than 98% of them have localisation error less than 0.3. We note that accurate localisation is not needed/done by the center sieve since we do not use an explicit shape model centred around it.

We compare the results using F2 and F3 in terms of localisation error of each individual facial point. The relative improvement of F3 in comparison to F2, that is defined as the error reduction over the original error, is shown in Fig. 2.24. There are four facial landmarks (the two eye brow and eye outer corners) with more than 40% relative improvement over the baseline (F2) in mean localisation error. Three facial points (right eye left corner, nose

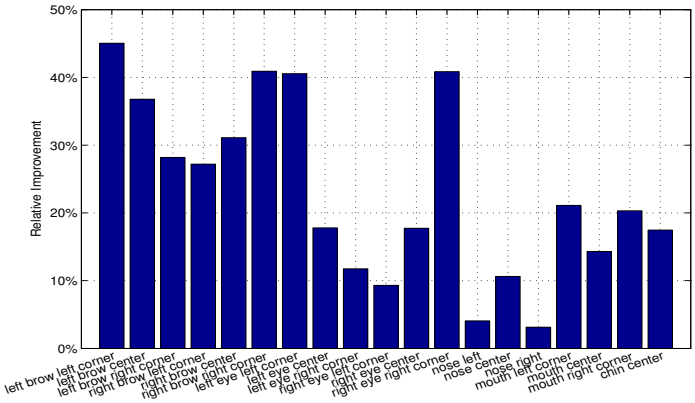
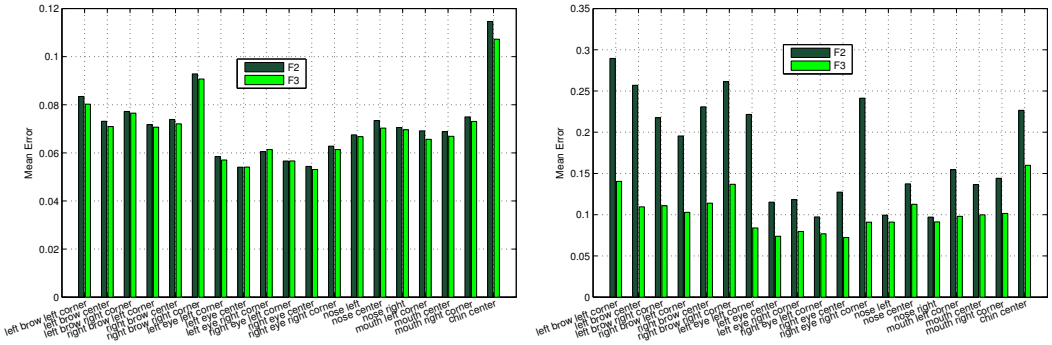


Fig. 2.24 Relative improvement by using the face center sieve.

left and nose right) show less than 10% relative improvement since these points are less frequently occluded and therefore easier to localize. In order to illustrate better the efficacy



(a) (b)

Fig. 2.25 Performance of sieves associated with the face center on the AFLW. The left and right are landmark-wise mean error results on AFLW_TestI and AFLW_TestII respectively. Note that the Y axis range of (b) is different from that of (a).

of the sieves on difficult images, we split the test set containing 1000 images into two sets, **AFLW_TestI** and **AFLW_TestII**, the former containing "easy" and the latter containing "difficult" images. We do so by applying the F2 detector (now regarded as a baseline) on the whole test dataset and putting into the AFLW_TestI the face images with average localisation error less than 0.1 (663 face images on average) and into the AFLW_TestII the rest (337 face images on average). We report results on them separately. As shown in Fig. 2.25a, in the easy set, applying the sieve only has very minor improvement, 2.3% in average. By contrast, in the difficult set, as shown in Fig. 2.25b, the improvement is very significant. The average relative improvement of the 19 points is 38% and the improvement is more significant for the difficult facial points, for instance the left eye brow left corner (51.5%, 0.1404

vs 0.2895), the right eye right corner (62.3% 0.0910 vs. 0.2413). The superior performance of F3 over F2 significantly validates the efficacy of our sieves, particularly on 'difficult' images.

Performance of votes aggregating The aggregating of the votes is controlled by a proximity threshold associated with individual facial landmark. In F3 we use threshold that are optimized for each facial landmark during training and in F4 reset the proximity threshold to infinity , that allows votes from the whole face region. The results are shown in Fig. 2.22. By taking all votes into account, F4 has the lowest performance for errors less than 0.1. Its distribution rises to a similar level to F3 and becomes higher than F2 at around 0.2 error. This shows that taking all votes into account leads to robustness but degrades the localisation accuracy. The efficacy of votes aggregating is best shown by comparing the results of F5 and that of F3. Even though we cannot observe a large margin of improvement in this figure, we note that the votes aggregating performed only when it is necessary, in most cases when heavy occlusion is present. In our four test experiments, the proportion of facial points with different aggregating steps (defined in Algorithm 4) is shown in Table 2.8, in total only 20% of them adjust the threshold to aggregate the votes.

Table 2.8 Aggregating steps proportion

Steps	$k=1$	$k=2$	$k=3$	Total
Percentage	10.5%	7.51%	2.43%	20.44%

We also evaluate the overall performance of using the sieves and aggregating by comparing F5 with F2. Though F2 has better performance than the plain forest F1 and is formalized as a type of our sieves, we treat it as the baseline method here because we want to highlight the original contribution of this work, as the idea of F2 originally proposed in (Sun et al., 2012) for human pose estimation. The improvement plot over the baseline error (CRF-S) is shown in Fig. 2.26, which validates that the improvement is correlated with the 'difficulty' level of the test images, i.e., our method produces large improvement when the baseline method has big error.

Landmark unreliability We qualitatively show some examples of facial landmarks unreliability detection in Fig. 2.27, where the number associated with each point location, that is also the aggregating step, intuitively reflect the unreliability level of the point. Since the unreliability of a region can be caused by several reasons, it is very difficult to determine it using low-level image feature. Our method model explores the information from middle level features, extracted from the voting map. We also note that some unreliable facial

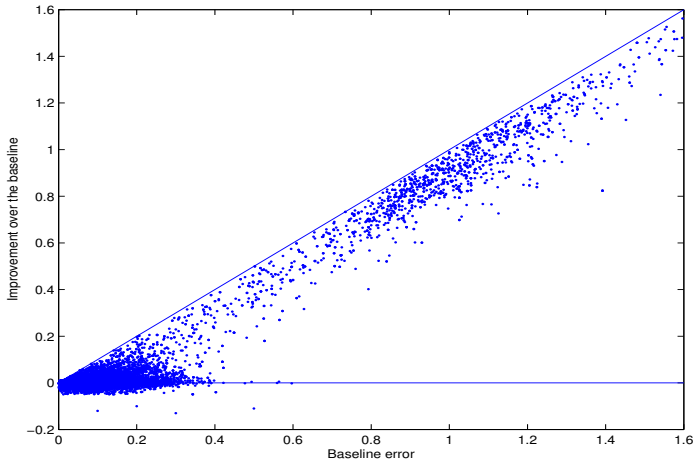


Fig. 2.26 Improvement plot over the baseline error (CRF-S).

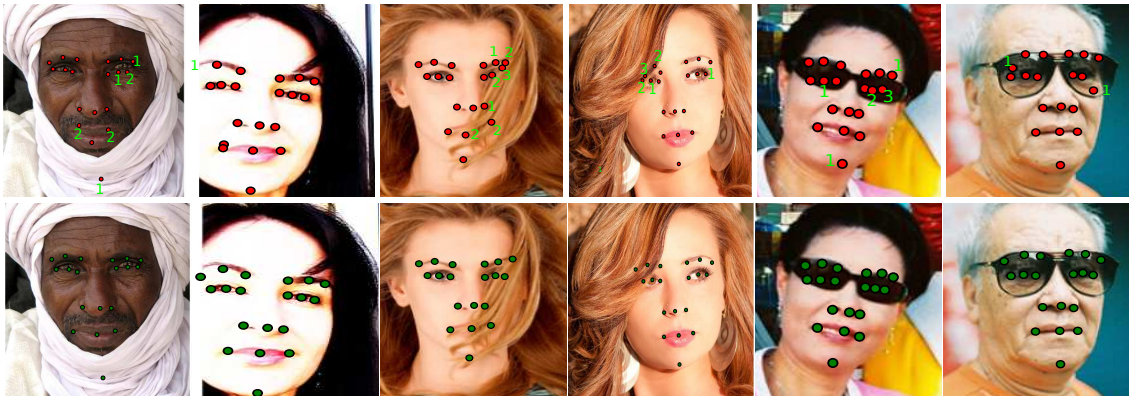


Fig. 2.27 Example results from the AFLW dataset before (top row) and after (bottom row) the votes aggregating. The value beside the red dot in the top row indicates the unreliability/ step length of aggregating. For clarity, the reliable point where no aggregating is needed, i.e. 0 is not shown in the figure.

points, like the eye corners under sunglasses in the last two columns, are not well identified. This is because during when training time of the OC-SVM, such images are used as positive training samples since the localisation accuracy is high. Thus further validates our sieving step is very robust to such kind of occlusion.

Face Alignment Comparison In this section we compare the performance of our proposed method with the existing face alignment approaches, namely the closely related random forests-based methods and other state-of-the-art methods. We do so on several widely used datasets.

Comparison with (Dantone et al., 2012b) on LFW A work that is closely related to

Fig. 2.28 Detection results of example images from LFW. The upper shows the results by CRF detector (Dantone et al., 2012b) and the lower shows the results of our method.

our method is the CRF proposed in (Dantone et al., 2012b). It reports the best performance on the LFW dataset. We evaluate the contribution of our sieve associated with latent continuous variable by comparing with its publicly available trained model. We randomly select 1000 images from the dataset for testing and split them into two sets, namely **LFW_TestI** and **LFW_TestII** according to the average localisation error of the CRF detector. In this way we create an 'easy' partition, namely the **LFW_TestI**, where the average point localisation error of the CRF is less than 0.1, and a 'difficult' partition, namely the **LFW_TestII**, where the average point localisation error of the CRF is larger than 0.1. We repeated this 4 times and on average 118 out of 1000 face images ended up into LFW_TestII. This small number is due to the fact that the face images in the LFW dataset are relatively easy. Only a few of them contain occlusions caused by head pose, hair and sunglasses. The absolute improvement of mean error and accuracy (using the definition of (Dantone et al., 2012b)) on the LFW_TestI and LFW_TestII are shown in Fig. 2.29 and Fig. 2.30. On LFW_TestI, there are some points our method even performs slightly worse, but the difference is negligible. To give the reader an idea, the maximum difference in the average point error is around 0.05 pixels. The maximum difference in the accuracy is also very small, namely around 0.5%. This is expected since our method is designed to maintain the performance of the baseline regression forests on "easy" images. On the contrary, the improvement on LFW_TestII is noticeable. The absolute reduction in the mean error for the *left eye left* point in average is around 0.4 pixels and that of the *right eye right point* is around 0.3 pixels. The differences on other points are not so noticeable. There are three points (*left eye left*, *left eye right* and *right eye right*) with more than 6% increase in detection accuracy.

As can be seen from the example images shown in Fig. 2.28, since the CRF detector (Dantone et al., 2012b) localizes each individual landmark in a completely independent

way, there are some points that are localized incorrectly due to occlusion or shadows caused by pose, hair or glasses. On the contrary, after applying our sieves associated with the face box center, based on the same trained model, our method is able to deal with the partial occlusion in an efficient way.

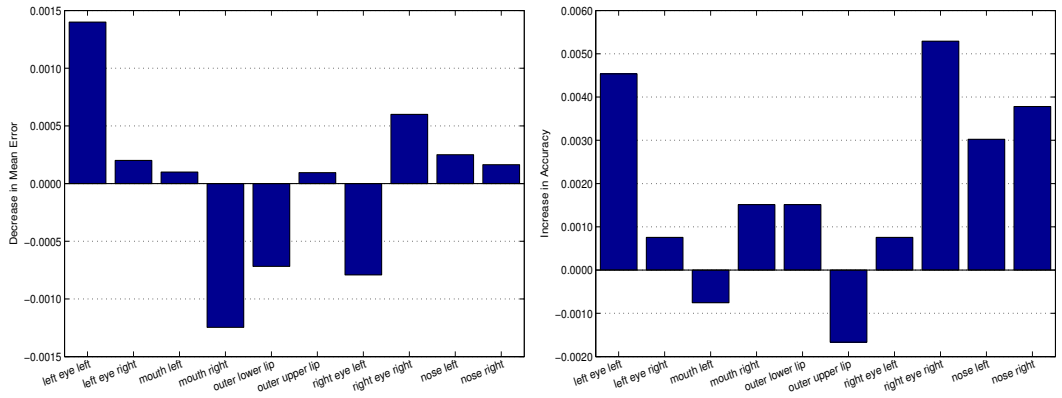


Fig. 2.29 Results on the LFW, compared to (Dantone et al., 2012b). The left and right are respectively the mean error decrease and accuracy increase on the LFW_TestI.

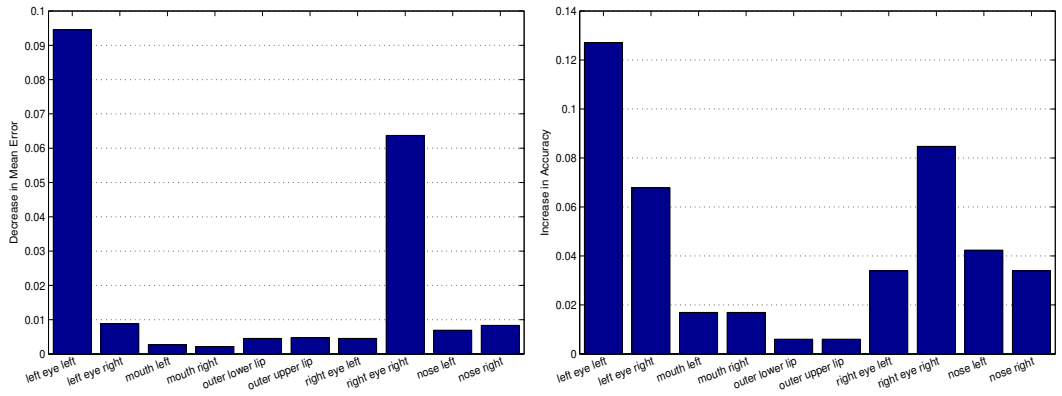
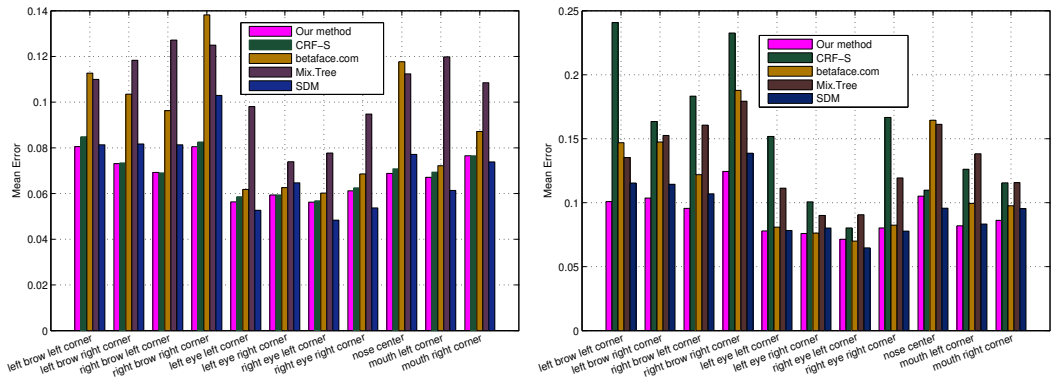


Fig. 2.30 Results on the LFW, compared to (Dantone et al., 2012b).The left and right are respectively the mean error decrease and accuracy increase on LFW_TestII Note that the range of the Y axis is different from that of Fig. 2.29.

Comparison on AFLW

We compare the overall performance of our proposed method with methods from the academic community as well as commercial systems, namely (1) the structured-output regression forests (SO-RF) in (Yang and Patras, 2012), (2) the regression forests based CLM (RF-CLM) (Cootes et al., 2012), (3) the mixture-of-trees (Mix.Tree) (Zhu and Ramanan, 2012), (4) Xiong and De la Torre's Supervised Descent Method (SDM) (Xiong and De la



(a) (b)

Fig. 2.31 Results of our method on AFLW_TestI (Left) and AFLW_TestII (Right), compared to (Sun et al., 2012; Xiong and De la Torre, 2013; Zhu and Ramanan, 2012) and betaface.com (Betaface).

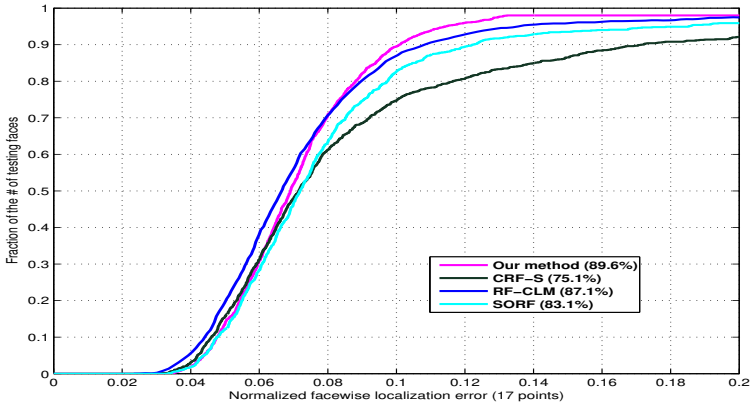


Fig. 2.32 Results of our method on the AFLW, compared to random forests-based methods (Cootes et al., 2012; Yang and Patras, 2012). The numbers in legend of (c) are the percentage of test faces that have average error below 10%.

Torre, 2013) and (5) betaface.com's face detection module (Betaface). Since betaface.com, Mix.Tree models and SDM detector embed face detection with landmarks detection, for fair comparison we build our algorithm on top of a Viola-Jones face detector from the Matlab computer vision toolbox. We manually discard missed or incorrect detections (e.g. sometimes Mix.Tree detected a half face) by any method when calculating the error. Among 1000 images, there are 74 missed face detections for betaface.com, 113 for Mix.Tree, 127 for SDM, and 89 for Matlab Viola-Jones detector. Though SDM also uses the Viola-Jones face detector (Viola and Jones, 2001) in OpenCV, the result is slightly worse than that provided Matlab toolbox, probably because different trained models are applied. Mix.Tree failed to detect small faces because they were trained on large faces where all landmarks are clearly visible. The test set then contains 776 images (555 in AFLW_TestI and 221 in AFLW_TestII). We compare results of 11 common points to CRF-S, betaface.com, SDM and Mix.Tree as shown in Fig. 2.31a and Fig. 2.31b. On the AFLW_TestI we see that both CRF-S and our method perform better than Mix.Tree and betaface.com, and slightly worse than SDM. On AFLW_TestII, CRF-S performs significantly worse while the other existing methods and our method have more stable performance. Our method performs better than Mix.Tree and betaface.com, and on par with SDM.

In Fig. 2.32 we compare the average localisation error of all the 17 internal points on a face (the chin center and mouth center are excluded) of our method with the random forests-based method, i.e., CRF-S, SO-RF and RF-CLM. We train SO-RF model on AFLW using the code provided by the authors using the same experimental setting as that of CRF-S and compare the reported result of RF-CLM. The markup of RF-CLM is slightly different since their results are for 17 points but two are annotated by the authors, that are not publicly available. As can be seen in Fig. 2.32, where the error cumulative distribution of the random forests-related methods is shown, our method performs on par with RF-CLM and significantly better than SORF and the CRF-S, though both RF-CLM and SORF are based on shape model fitting. An example image is shown in Fig. 2.33 where our method performs better than not only the local detection method, like CR-S, but also the ones using shape models such as (Yang and Patras, 2012; Zhu and Ramanan, 2012). In addition, we have found that in terms of computational complexity and in terms of how well it deals with low quality images, our method performs considerably better than the Mix.Tree model. However, as shown in Fig. 2.34, unlike the Mix.Tree, our method fails on side view faces since we have not used such images in training.

Car Alignment Comparison We evaluate our method on car alignment using the same experimental set-up presented in (Boddeti et al., 2013). More specifically, for each view,

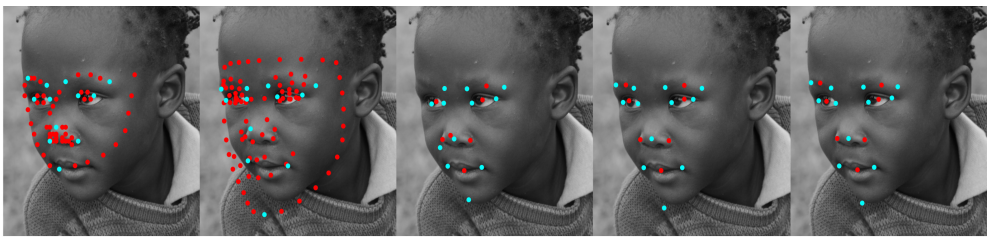


Fig. 2.33 Left to right: Results for Mix.Tree, betaface.com, CRF-S, SO-RF and our method on an image from AFLW. The blue dots are the 12 common points.

Fig. 2.34 An example image from AFW (Zhu and Ramanan, 2012) with results from Mix.Tree (Left) and our method (Right)

the landmark-wise average RMSE over four different subsets is reported. More precisely, 1) the average over all images, 2) the average over images with occluded landmarks, 3) the average over the unoccluded landmarks in partially occluded images and 4) the average over the occluded landmarks in partially occluded images. The results are shown Fig. 2.35. From top to bottom are the results of the four different subsets respectively and from left to right are the results for views from 2 to 4. The front and back view images are less challenging and their results are not shown here.

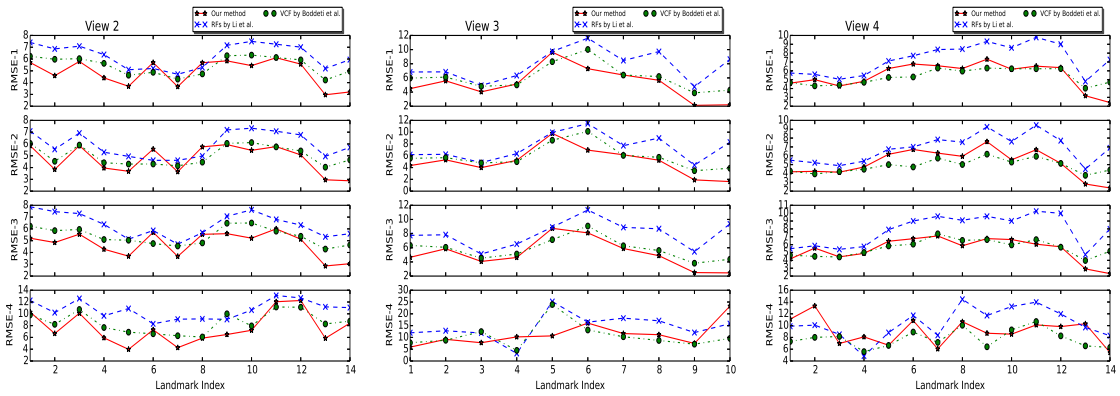


Fig. 2.35 Landmark wise RMSE error for each view, from top to bottom: 1) all image, 2) images with no occlusions, 3) unoccluded landmarks of partially occluded image, 4) occluded landmarks of partially occluded image.

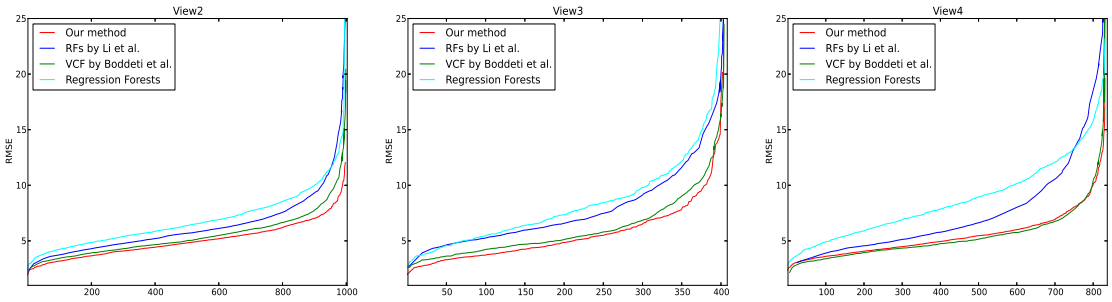


Fig. 2.36 Comparison of the sorted RMSE for each view to the VCF model in (Boddeti et al., 2013), random forests model in (Li et al., 2011) and the baseline Regression Forests in our work.

.

We compare the baseline regression forests and two other methods, the Random Forests (RFs) based method proposed by Li et al. (Li et al., 2011) and the Vector Correlation Filter (VCF) method by Boddeti et al. (Boddeti et al., 2013). We compare with their best reported results (Boddeti et al., 2013), i.e. the results from RFs with RANSAC BPSI shape model and VCF with Greedy BPSI shape model (see (Boddeti et al., 2013) better for details). We observe that our method is able to align most of the landmarks in the lower RMSE for different subsets of view2 and view3. For view4, our method performs better than the RFs-based method and on par with the VCF method. To further investigate the error distribution we also compare the individually sorted errors for each view in Fig. 2.36. We observe that in view2 and view3, our method performs significantly better, i.e., for a given error tolerance our method aligns more images compared to state-of-the-art methods while the baseline regression forests-based method performs worse. In view4, our method performs better than RFs and similar to VCFs. The superior performance over the baseline plain regression forests validate the efficacy of our proposed votes sieving and automatic aggregating. Example results from all the five views are shown in Fig 2.37. The top row shows the results from the plain regression forests, that is unable to handle occlusions. The bottom row shows the results of our method.

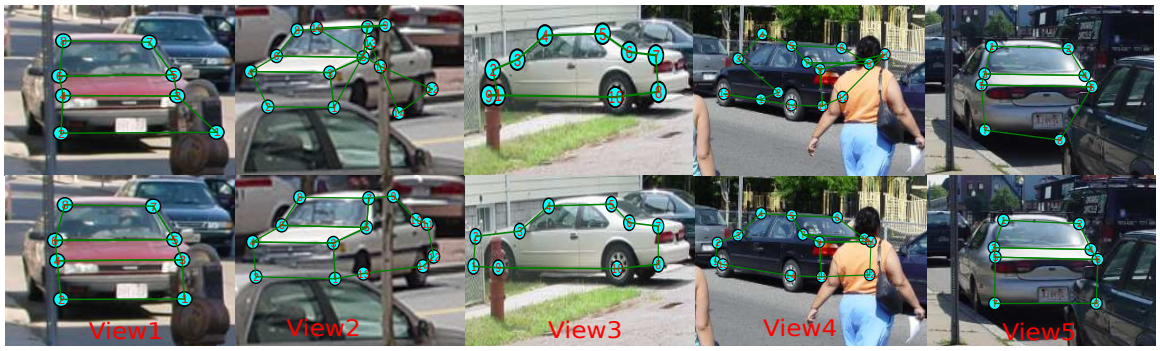


Fig. 2.37 Detection results of example images of different views from CMU-CW. The upper shows the results by plain regression forests and the lower shows the results of our method.

2.4 Summary

In this chapter, we first focus on the training stage and present local based methods for face alignment using a regression forest framework. We show that privileged information such as head poses at the training stage can be used for building better decision trees and constructing conditional models. It is useful to deal with head variations. We also show that learning structured output models on the leaf nodes is able to deal with partial occlusions. The proposed method is different from the traditional methods as it does not require any additional shape models. On the contrary, it incorporates structure information within the regression forests and demonstrates better or similar performance to other regression forest based methods, with or without additional explicit shape models.

We then focus on the testing stage and propose a scheme that fine tunes the regression forest votes. Before accumulating the votes to a Hough map for detection, this scheme filters out the false positives votes by using sieves which impose agreement on latent discrete or continuous variables. In addition, it proposes a votes aggregating strategy which automatically seeks additional votes when necessary. The proposed method is validated on two challenging tasks: facial feature detection and car alignment. It yields performance superior or close to the state-of-the-art on the most challenging datasets with images collected in the wild.

As a local based method, regression forest based methods carry out discriminative local detection in a very efficient and effective fashion. Despite no explicit shape model being built, original regression forest can obtain competitive results. However, it also inherits some drawbacks of the local based methods, for instance the computational complexity increases sharply as the number of desired facial landmarks gets larger. Also, regression forest approaches do not perform better when comparing to the recent holistic-based face alignment methods. Nevertheless, this line of methods follow very different setting and the

advantage of local based methods will be useful in certain circumstance for instance when the initialization is not reliable for cascaded holistic method. Further more, it has very good generalization capacity and can be applied to other object alignment problems, not only human faces and the car alignment example we have shown.

Chapter 3

Holistic based Face Alignment

In previous chapter, we presented local based regression forest methods for face alignment. They show competitive performance on face images collected from unconstrained conditions. However, there are some inherited drawbacks of the local based methods. A critical one is that the model and computational complexity is high, due to the fact that each landmarks should be detected separately and the global constraint (like our votes fine-tuning or traditional CLM) is usually applied as an ad hoc step. The efficiency becomes problematic especially when a large number of facial landmarks are involved.

In this chapter, we focus on holistic approaches for efficient face alignment of a large number of facial landmarks. In contrast to the local based methods, holistic method regresses the face shape as a whole thus the number of facial landmarks has little impact on the model size and computational complexity. It starts from a raw shape and updates it in a coarse to fine manner. We present three holistic methods, i.e., the cascaded forests, the random subspace descent method and an adaptation of the Cascaded Pose Regression method for multi-modality (RGB and sketch image) face alignment. In Section 3.1 we briefly review the framework of cascaded pose regression. In Section 3.2 we present our method of a cascade of forests for face alignment, which uses a Regression Forest as the primitive regressor in the CPR framework built on raw pixel features. In Section 3.3, we present Random Subspace Supervised Descent Method (RSSDM). It improves the standard Supervised Descent Method (SDM) (Xiong and De la Torre, 2014) and uses hand-crafted HOG features.

3.1 General Framework of Cascaded Pose Regression

In this section, we briefly review the general framework of cascaded pose regression (CPR), based on the description of (Dollár et al., 2010).

The shape of an object (e.g. face) is often represented as a vector of landmark locations, i.e., $S = (y_1, ..., y_k, ..., y_K) \in \mathbf{R}^{2K}$, where K is the number of landmarks. $y_k \in \mathbf{R}^2$ is the 2D coordinates of the k-th landmark. CPR is formed by a cascade of T regressors, $R^{1...T}$. Shape estimation starts from an initial shape S^0 and progressively refines the pose. Each regressor refines the pose by producing an update, ΔS, which is added up to the current shape estimate, that is,

$$S^t = S^{t-1} + \Delta S. \tag{3.1}$$

The update ΔS is returned by the regressor that takes the previous pose estimation and the image feature I as inputs:

$$\Delta S = R^t(S^{t-1}, I) \tag{3.2}$$

An important aspect that differentiates this CPR framework from the classic boosted approaches is the feature re-sampling process. More specifically, instead of using the fixed features, the input feature for regressor R^t is calculated relative to the current pose estimation. This is often called a pose-indexed feature as in (Dollár et al., 2010). This introduces weak geometric invariance into the cascade process and shows good performance in practice. The CPR is summarized in Algorithm 2 (Dollár et al., 2010).

Algorithm 2 Cascaded Pose Regression

Input: Image I, initial pose S^0
Output: Estimated pose S^T

1: **for** t=1 to T **do**
2: $f^t = h^t(I, S^{t-1})$ ▷ Shaped-indexed features
3: $\Delta S = R^t(f^t)$ ▷ Apply regressor R^t
4: $S^t = S^{t-1} + \Delta S$ ▷ update pose
5: **end for**

The above scheme holds several advantages. First, though for each stage, the pose-indexed feature is re-calculated, the original image feature, that can be more than image gray scale values, requires only one computation as a preprocessing step. Thus the feature re-calculation in practice is highly efficient. Second, the number of the landmarks representing the object shape has little impact on the testing efficiency since it only involves a vector addition operation, while other methods like (Dantone et al., 2012b; Saragih and Goecke, 2007; Zhu and Ramanan, 2012), the computational complexity is linearly or exponentially related to the number of landmarks. Thus, besides the effectiveness in real application, the CPR is very popular due to its computational efficiency.

Cascaded methods differ from each other in two respects, i.e., what type of regression models are used (i.e. R^t) and what type of features are extracted (i.e. h^t), as we will present

in the following sections.

3.2 Cascaded Forests for Face Alignment

In this section we will present our method of cascaded forests for face alignment. We follow the main scheme of CPR and use Regression Forests as our primitive regressor at each stage, given the previous work of regression forests. We also propose an intelligent initialization scheme that can be used for the general CPR framework.

3.2.1 CPR training

In order to train a cascade of forests, let us assume we are given a set of n training samples $\{(I_i, S_i)\}_{i=1}^{n}$. I_i represents the image of the i sample and S_i, the ground truth shape. We assume here that the image only contains the face or has the bounding box of the face, since our algorithm is built on top of the face detection. For each training sample, we randomly select 20 ground truth poses from the training set except its own. We treat an individual training sample with a different initialization as a new sample. Each training sample is now represented by a triplet, that is $(I_i, S_i, \bar{S}_i)$, with $\bar{S}_i$ the initial pose. The augmented number of training samples is therefore $N = 20 \times n$.

For each training sample, with the current pose $\bar{S}$ and the ground truth pose S, the target update vector the regressor aims to estimate is

$$\Delta S = S - \bar{S}. \tag{3.3}$$

Thus at each stage we train a regressor at each stage that minimizes the square error loss, given the features f_i^t calculated using the previous pose state.

$$R^t = \arg\min_R \sum_i |R(f_i^t) - \Delta S_i^t|^2 \tag{3.4}$$

The training procedure of the CPR is summarized in Algorithm 3.

3.2.2 Forest-based regressor

In this section we discuss how as a primitive regressor, a forest is trained. A forest is an ensemble of regression trees. The simplest version of a forest consists of one tree. Thus we first discuss how a regression tree is trained and then discuss the ensemble method. Let $\mathcal{X}$ denote the input space, $\mathcal{Y}$ the output space. Each tree is induced based on a randomly selected subset of the training data $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. An empty tree starts with only one root node. Then a number of *split test function* candidates $\phi : \mathcal{X} \to \{0, 1\}$ are generated, which determines whether to route a data sample $x \in \mathcal{X}$ reaching it to go left or right child. $\mathcal{P}_{left}(\phi)$

Algorithm 3 Cascaded Pose Regression Training

Input: training data $(I_i, S_i, \bar{S}_i)$ for $i = 1...N$
Output: $R = (R^1, ...R^T)$

for t=1 to T **do**
  for all $i \in (1...N)$ **do**
    $\Delta S_i^t = S_i^t - \bar{S}_i^t$ ▷ Calculate ΔS_i^t
    $f_i^t = h^t(I_i, \bar{S}_i^{t-1})$ ▷ Shaped-indexed features
  end for
  $R^t = \arg\min_R \sum_i |R(f_i^t) - \Delta S_i^t|$
  for all $i \in (1...N)$ **do**
    $\bar{S}_i^t := \bar{S}_i^t + R(f_i^t)$ ▷ Update current pose
  end for
end for

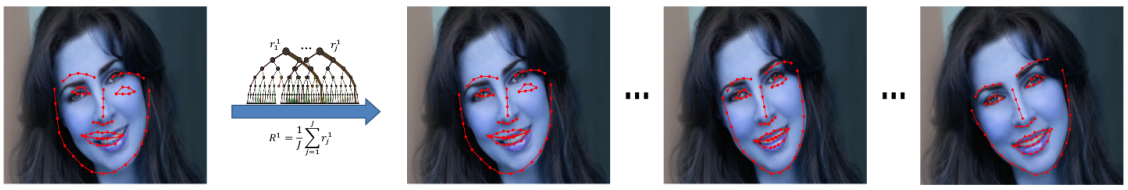


Fig. 3.1 Starting from a raw pose, our method refines the face shape recursively by using different stages of regression forests, organised in a cascade.

and $\mathcal{P}_{right}(\phi)$. According to one specific split function ϕ, the set of data, denoted by $\mathcal{P}$ ($\mathcal{P} \subseteq \mathcal{D}$), at the node will be partitioned into two, $\mathcal{P}_{left}(\phi)$ and $\mathcal{P}_{right}(\phi)$. Based on the partition, each candidate split function is evaluated according to a certain loss function, so that the best split function, that is the one with the minimum value of the loss function, is selected, i.e. $\phi^* = \arg\min_\phi \mathcal{L}(\phi)$. The node is parametrized by the selected split function ϕ^*. Then, the training set is partitioned according to this split function into two subsets that are propagated to the two child nodes. The same procedure is recursively applied at each subsequent child node. The procedure stops and a leaf node is created when certain criteria is met, typically, when there are fewer than a minimum number of training data or a maximum tree depth is reached. At each leaf node, a regression model is learned and stored.

According to the above description of tree construction, aside from the macro parameters of the tree, there are two tasks involved: specifying the split test function at each internal node and learning the regression model at each leaf node. As discussed before, in order the keep the high efficiency of the algorithm, we focus on very simple test functions, that is to compare the feature values at two pixel locations. Besides gray scale, other pixel-wise features can also be used such as the Gabor features, with an additional cost of feature computation. So as to generate a pool of split testing functions, we randomly select two landmark numbers, l_1 and l_2. Then we generate an random offset to each of the two landmark locations, δ_1 and δ_2. Thus for the training sample i the first location feature indexed by the current pose is:

$$x_i^1 = I_i^{\bar{S}_i(l_1)+\delta_1} \tag{3.5}$$

where $\bar{S}_i(l_1)$ denotes the image location of the l_1-th landmark, deduced by the current pose estimate $\bar{S}_i$. The second location feature is $x_i^2 = I_i^{\bar{S}_i(l_2)+\delta_2}$. The split function consists of five parameters, $\phi = (l_1, l_2, \delta_1, \delta_2, \tau)$, where τ is a threshold variable. Formally the split function ϕ is written as:

$$\phi_{(l_1,l_2,\delta_1,\delta_2,\tau)}(I_i, \bar{S}_i) = \begin{cases} 0 & \text{if} \quad x_i^1 - x_i^2 > \tau \\ 1 & \text{otherwise} \end{cases} \tag{3.6}$$

In order to select the best split function candidate at each node, based on the loss function in Eq. 3.4, we rewrite the objective function as:

$$\mathcal{L}(\mathcal{P}, \phi) = \sum_{c \in \{left, right\}} \sum_{i \in \mathcal{P}_c} |\Delta S_i - \mu_c| \tag{3.7}$$

where

$$\mu_c = \frac{1}{|\mathcal{P}_c|} \sum_{i \in \mathcal{P}_c} \Delta S_i \tag{3.8}$$

is the mean value of the update vectors. The optimal split candidate is selected as the one

which has minimized the above loss function, i.e.,

$$\phi^* = \arg\min_{\phi} \mathcal{L}(\mathcal{P}, \phi) \tag{3.9}$$

When training samples arrive the leaf node, the regression model is calculated as the average pose update vector of all the training samples in question, similar to Eq. (3.8).

Instead of using one tree as weak regressor at each stage of the cascade as described above, we train a forest consisting of a set of trees, that is $R = \{r_j\}_{j=1}^{J}$. The output of the forest is the average of the predictions of all the trees, that is,

$$R^t(S^{t-1}, I) = \frac{1}{J}\sum_{j=1}^{J} r_j^t(S^{t-1}, I). \tag{3.10}$$

The averaging regularization is able to deal with the general over-fitting problem in boosting regression. This will be demonstrated in the experiments.

3.2.3 Intelligent initialization

The output of CPR is initialization dependent and very sensitive to bad initializations. Previous approaches such as (Cao et al., 2012; Dollár et al., 2010) propose to run multiple different initializations and pick up the median of all the predictions as the final output. Each initialization is treated in a completely independent way until the output is calculated. The theoretical support of selecting the median value is not well understood. Also there is no guidance on how to choose the multiple initializations.

We propose an intelligent initialization scheme, which works in a coarse-to-fine manner. We build an initialization pose dataset with M instances, each with a unique pose consisting of K landmark locations. Given a testing image, we randomly select m initializations, $m \leq M$. The number of m is set to a large number, around ten times larger than the number of initializations used in the previous approaches (Cao et al., 2012; Dollár et al., 2010). Instead of applying the whole cascade on the m initializations, we apply only a few top stages of the cascade and analyse their results. Specifically, we apply the mean-shift algorithm to find the mode of the estimated shapes using the small number of top stages, that is the shape with highest density in the shape space. Then the remaining cascade is applied on m' poses, which are closest to the shape mode. $m' \ll m$ is a very small number that can be even smaller than the initialization number in (Cao et al., 2012; Dollár et al., 2010).

We now discuss the theoretical support of this scheme. As discussed in (Cao et al., 2012), at the early stage, the regressors in the cascade aim at adjusting the global shape updates such as yaw, roll and scaling. In later stages, the regressors are dominated by the subtle

variations such as motions on eyes and lips. Therefore, we assume that a good initialization aligns the rough shape in a few stages while a bad initialization progresses towards a wrong position. Also we assume that in most of the cases, there are more good initializations than bad initializations given the fact that we have augmented multiple random initializations during the training stage. The first assumption is validated by the Principal Components analysis in (Cao et al., 2012) and the second assumption was implicitly used in the previous approaches. Given these two assumptions, we believe that the m' initializations we selected as discussed above are more reliable and are more likely to converge towards the correct pose position.

Since we only apply a very small number of stages in the cascade on the m raw initializations thus we can still expect very high evaluation efficiency. When m' initializations arrive the end of the cascade, since the number of $m's$ is very small, we calculate the distance between each pair and then select the the pair with minimum distance. The final output is calculated as the mean value of the selected pair, this is different from selecting the median value.

3.2.4 Experiment Setting

To evaluate the efficacy of the proposed approach, we conduct the experiments on face alignment from a single image. The face images are collected in uncontrolled environments, and taken from various viewpoints and often present in cluttered backgrounds, with severe partial occlusion.

In this work we mainly focus the comparison on the LFPW and HELEN datasets, with the annotation from 300-W, as it provides annotations of large number of common landmarks for several widely used datasets. Since 300-W has not made its test images publicly available, we follow the experimental setting (training/testing partition) of LFPW and HELEN when comparing to other methods. We compare the Regression Forests related methods on LFW dataset and follow the experiment setting of (Dantone et al., 2012b).

Implementation details

We train our model using the training partition of LFPW and HELEN with 68 landmark annotations provided by 300-W. As mentioned in Section 3.2.1, we augment the training data with 20 training poses for each training sample. For each tree in the forest, we keep the same parameter setting. The depth of the tree is set to 5. At each internal node, in order to select the best split function, we generate 400 candidate split functions that consists of a pair of locations, the corresponding offsets as well as a threshold. In the cascade, at each

stage, i.e. for each forest we use 5 weak tree regressors and in total we have trained $T = 500$ stages of forests.

During testing, we create an initialization set with 500 pose instances, i.e. $M = 500$. In order to generate intelligent initializations, we set $m = 100$, i.e. randomly select 100 pose instances from M. We apply the top $\frac{1}{10}$ of the cascade on the m initialization instances and then select the best $m' = 5$ pose instances, as discussed in Section 3.2.3, that are allowed to go through remaining cascade and generate the final output.

Evaluation measurements

In the literature, it is commonly accepted that the individual detection error is measured as the distance between the detected landmark location and the ground truth, normalized as a fraction of the inter-ocular distance (Burgos-Artizzu et al., 2013; Cao et al., 2012; Dantone et al., 2012b; Valstar et al., 2010) (or the face size (Zhu and Ramanan, 2012)). In order to measure the performance on a dataset, there are several measurements proposed, including overall average landmarks error (Burgos-Artizzu et al., 2013), landmark-wise average error(Belhumeur et al., 2011; Dantone et al., 2012b), cumulative distribution function (CDF) of landmark-wise error (Cao et al., 2012; Dantone et al., 2012b), CDF of face-wise error (Cristinacce and Cootes, 2006) and failure rate (Burgos-Artizzu et al., 2013; Dantone et al., 2012b). As most of the current methods have achieved very high accuracy, within an error level of 10 (as a fraction), it is difficult to evaluate the algorithm using the CDF as most of the errors are within small values. For comparison, we report the the overall and landmark-wise average error as well as the failure rate of the algorithm. The failure is determined if the average error is larger than 10, as defined in (Dantone et al., 2012b).

3.2.5 Results

Method evaluation

Cascade stages It is an open question that how many stages in the cascade should be set for a specific problem. Since the testing time just depends linearly on the number of stages in the cascade, increasing the number of the stages does not influence the testing much. We have tried in our experiments by increasing the number from 100 to 450, with a step size of 50. The performances on the LFPW and HELEN are shown in Fig. 3.3 and Fig. 3.4 respectively. Note that the failure proportion here is calculated as when the face wise average error over all 68 landmarks is larger than 10 (as a fraction), that is different from Table 3.3, where the failure is calculated over the common 49 points. On testing images from both datasets, the mean error and failure proportions decrease gradually while the

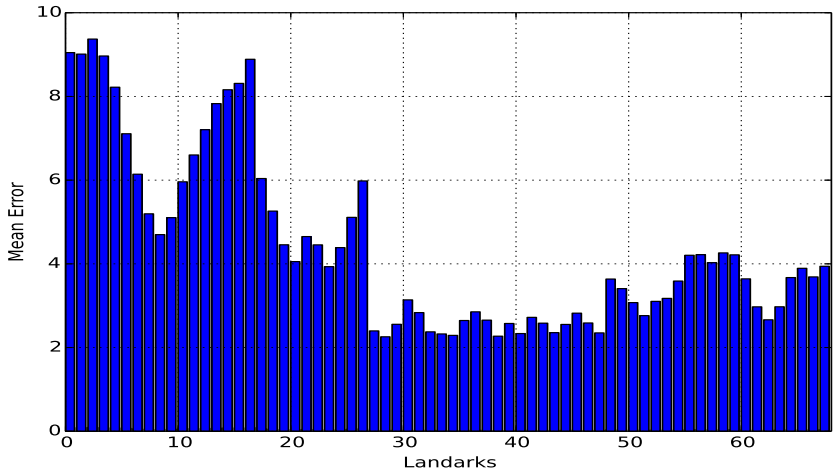


Fig. 3.2 Mean error of individual landmarks on the HELEN.

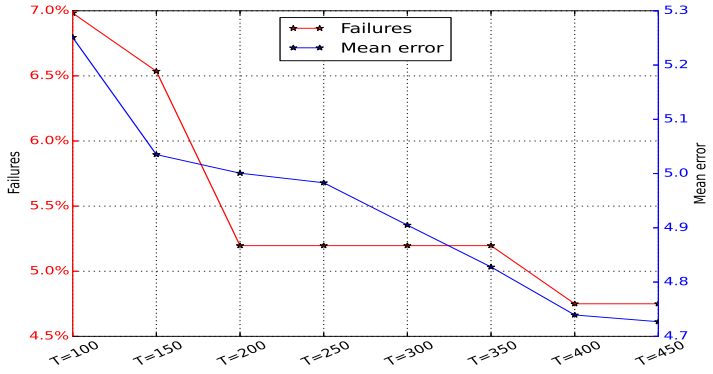


Fig. 3.3 Performance against cascade levels on LFPW.

number of stages increases from 100 to 400. On the HELEN dataset, the failure percentage decreases from around 11.0% to 7.7% and the mean error decreases from 5.2 to 4.75. On the LFPW dataset, The failure percentage decreases from 6.8% to 4.78% and the mean error decreases from 5.5 to 4.78. When the stage number keeps increasing, on the HELEN dataset, the performance decreases while on the LFPW dataset, the performance has slight change. Thus we will set the $T = 400$ as the optimized stage number in the cascade. The overall performance of landmark-wise mean error using 400 cascade stages regression on the HELEN is shown in Fig. 3.2, where the landmark IDs are defined in Fig. 1.7. All the landmarks mean errors are smaller than 10, and the error of the landmarks along the face contours (from 1 to 17) are bigger than the internal landmarks.

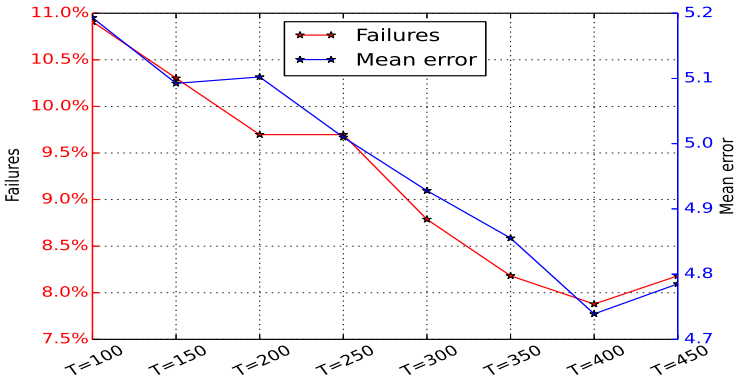


Fig. 3.4 Performance against cascade levels on HELEN.

Intelligent initialization

In this section we evaluate the effectiveness of the proposed intelligent initialization scheme. We compare it to the blind initialization scheme that is used in the traditional CPR method, i.e. to propagate a set of initializations till the final stage of the cascade. To make a fair comparison, instead of selecting the median values, we also apply our proposed method to calculate the final shape pose. The comparison is shown in Table 3.1. As can be seen, using the intelligent initialization just slightly reduces the mean error, but greatly reduces the failures. Note that the failures and average landmark error is calculated over all 68 landmarks.

Table 3.1 Intelligent initialization vs. blind initialization.

	Blind Initialization	Intelligent Initialization
LFPW	8.1%/4.95	4.8%/4.73
HELEN	10.2%/5.19	7.7%/4.78

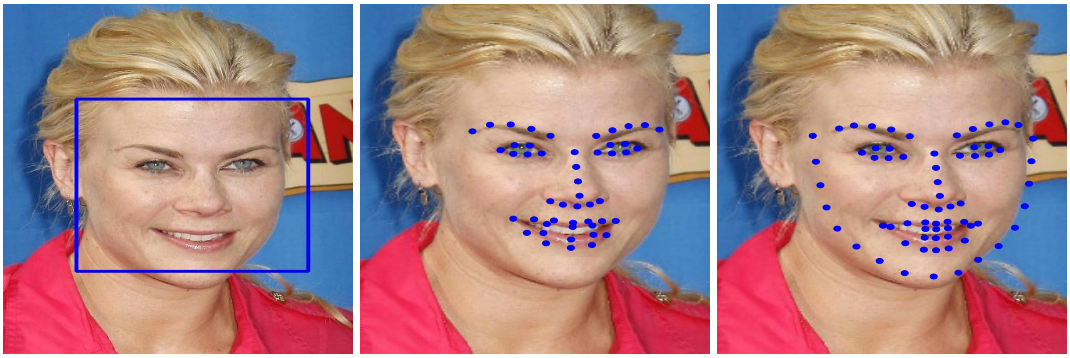
Image feature As we have discussed in Section 3.2.2, in the primitive regressor, not only the gray scale image feature can be used. We also evaluate other high level grid based features like the Gobor feature, image edges, etc. In order to train the model with the compact features, we set the training parameters the same as that is used to train the model with single channel gray scale feature. When testing on the same images, the model with compact features (gray scale, 32 channels of Gabor features and two channels of gradient) performs slightly better in terms of the failures. More specifically, it reduces the failures by 1.7% on the LFPW and 1.3% on the HELEN. In terms of alignment accuracy, there is no

significant difference by using the compact features. However, due to the consuming Gabor feature computation, the speed (FPS) is 3 times slower. Therefore in order to keep the highly computational efficiency, we will only use the gray scale feature in our experiments.

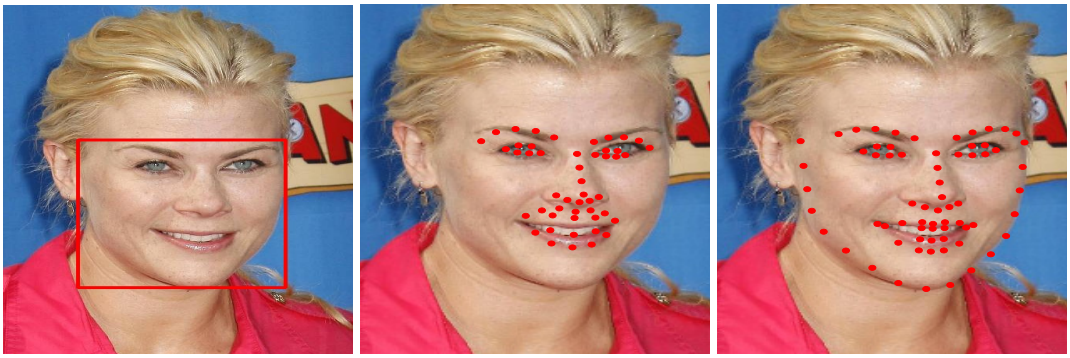
Different face detection Most approaches in the face alignment (facial feature detection) assume the face detection (face bounding box) is available. Only a few methods like (Zhu and Ramanan, 2012) integrates the face detection and landmark detection. However, there is no standard definition of the face bounding box. It varies from methods. The most commonly used face detection method is the Viola-Jones face detector (Viola and Jones, 2001). In other face databases, different face bounding boxes are provided like the 300-W. Since all boosting method starts from an initial shape, the face bounding box affects the initialization shape, which in return affects the final shape regression. We compare our method to Xiong and De la Torre's Supervised Descent Method (**SDM**) (Xiong and De la Torre, 2013), with different face detection. An example face image from the LFPW is shown in Fig. 3.5. The face bounding box returned from the Viola-Jones detector is Fig. 3.5a and the face bounding box in 300-W is shown in Fig. 3.5d. The facial landmarks detected from the SDM and our method with the two different face detections are shown in the second the third column respectively. Fig. 3.5b shows the landmarks detection based on the Viola-Jones face bounding box while Fig. 3.5e shows the landmarks detection based on face bounding box in the 300-W. Fig. 3.5c and Fig. 3.5f are the landmarks detection results of our method based on the Viola-Jones detector and 300-w face detector. Since the SDM is trained on the face images with bounding boxes returned from Viola-Jones detector, the landmark localisation in Fig. 3.5b is much more accurate than that in Fig. 3.5e. On the testing images, by using a different face detector, the failure rate of SDM increases by 21% while that of our method increases 7%, 1/3 of SDM. This validates our method is more robust to different face detection initialization.

Comparison to regression forest methods

Since our method uses Regression Forest (RF) as the primitive regressor, we first compare the related methods that use RF for facial feature detection including the Conditional Regression Forests (C-RF) method in (Dantone et al., 2012b), the Regression Forest based Constrained Local Model (RF-CLM) in (Cootes et al., 2012) and the recent Regression Forests votes sieving (RF-S) in Section 2.3. We note that the RF in these methods is used in a different way from our proposed method. While in their RF framework, local patches are used to cast votes for individual landmarks, in our method, RF is used as a holistic regressor for the update of the whole shape. The comparison is made on the LFW dataset on



(a) Viola-Jones detector (b) SDM (c) Our method



(d) IBug Detector (e) SDM (f) Our method

Fig. 3.5 With different face detection initialization.

Table 3.2 Comparison to RF methods on LFW.

Methods	C-RF	RF-CLM	RF-S	Our method
Mean Error	7.1	6.5	6.2	5.3
Speed (FPS)	25	12	10	35

which the related methods reported results. We follow the experiment setting of (Dantone et al., 2012b) for all these methods. The results of the mean error of the 10 facial landmarks and the test run-time performance (It is measured on a standard 3.3GHz four-core machine) is shown in Table 3.2. Our RF-based approach outperforms the counterparts significantly in both accuracy and efficiency, despite the fact that the other RF methods use the four cores for parallel computation but our method uses only one core. The other RF methods work in a sliding window fashion and cast votes for each individual landmark separately, therefore the computational complexity grows exponentially when the number of landmarks increases. On the contrary, our method treats the shape as whole, thus the number of landmarks will not affect the run-time performance. Our method can also detect 68 landmarks on other datasets at a speed of 35FPS.

Comparison to other methods

Table 3.3 Comparison with the existing methods. C. represents the common 49 facial landmarks that SDM and other methods can detect while 66P represents the 66 common landmarks the methods except SDM can detect.

Method Description			LFPW			HELEN		
Method	Model trained on	# of points	C. ME	C. Fails	66P ME	C. ME	C. Fails	66P ME
Mix.Tree (Zhu and Ramanan, 2012)	Multi-PIE	68	11.4	27.3%	15.2	12.6	26%	14.7
DRMF (Asthana et al., 2013)	Multi-PIE+LFPW	66	4.4	7%	5.8	4.6	**4.8%**	5.4
SDM (Xiong and De la Torre, 2013)	Multi-PIE and LFW-A_C	49	4.27	**2.7%**	N/A	3.67	5.33%	N/A
CPR (Cao et al., 2012)	LFPW/HELEN	68	5.1	6.5%	5.7	4.8	7.5%	5.8
RCPR (Burgos-Artizzu et al., 2013)	LFPW/HELEN	68	4.9	4.2%	5.2	4.5	6.1%	5.2
Our method	LFPW/HELEN	68	**3.92**	3.5%	**4.91**	**3.65**	6.37%	**4.78**

Closely related to our work are the CPR-based methods (Burgos-Artizzu et al., 2013; Cao et al., 2012; Dollár et al., 2010; Efraty et al., 2011). The current one with the best performance is (Burgos-Artizzu et al., 2013), that has used additional occlusion annotation for model training. We use the code that is provided by (Burgos-Artizzu et al., 2013), which also contains a re-implementation of (Cao et al., 2012). We use the same training/testing setting as our model on the LFPW and HELEN dataset. The comparison is shown in Table 3.3. As can be seen, the proposed approach outperforms the baseline CPR model as well as the RCPR method. Note that, since there is no occlusion annotation on HELEN and

LFPW, we only use their feature extraction and their proposed smart restart components for a fair comparison. The superior performance validates the efficacy of our proposed strategy.

We also compare the performance of our approach with the state of the art methods with publicly available code. We compare with the following methods, 1) Xiong and De la Torre's Supervised Descent Method (**SDM**) (Xiong and De la Torre, 2013), 2) Asthana et al.'s Discriminative Response Map Fitting (**DRMF**) method (Asthana et al., 2013) running on the best performing tree-based model, 3) Zhu and Ramanan's Mixture of Trees (**Mix.Tree**) model (Zhu and Ramanan, 2012).

We apply the publicly available code of SDM, DRMF and Mix.Tree on the testing images from LFPW and HELEN in 300-W. From the description of the papers, the model of SDM detector is trained on Multi-PIE (Gross et al., 2010) and LFW-A&C datasets, DRMF, trained on Multi-PIE and the LFPW training set while Mix.Tree is trained on Multi-PIE. The CMU Multi-PIE face database contains more than 750,000 images of 337 people under various view points (15) and different illumination conditions while displaying a range of facial expressions. However it is not freely available to the public. Therefore, our model is trained on the freely available database in order to make the future comparison more convenient. All methods except the Mix.Tree and DRMF are built on top of the face detection. SDM is based on the Viola-Jones face detector while our method is based on the face detector in (Sagonas et al., 2013a). Thus for fair comparison, in case a face detector fails, we will manually set a proper bounding box for a face.

The comparison on the testing images from LFPW and HELEN is shown in Table 3.3. By comparing the mean error of the common 49 landmarks (C. ME) and the failures of common landmarks (C. Fails), we can clearly see the superior performance of our method over the Mix.Tree and the DRMF method, however the DRMF method performs best in terms of C. Fails on the HELEN. The failures of our method on the LFPW are just 10% of Mix.Tree and half of DRMF, which is a very significant improvement. Our method has comparable performance to the SDM (Xiong and De la Torre, 2013). The SDM has fewer failures while our method performs slightly better in terms of mean error on both databases. We also note the models used by these three existing methods were all trained on a large number of highly reliable face images from the Muli-PIE face database while the model of our method was trained on only a few thousands of face images. It shows superior or comparable performance when compared to the existing state of the art methods. One example image from each dataset is shown in Fig. 3.6. As can be seem, the Mix.Tree and DRMF both have difficulty to deal with the subtle variations caused by the facial expression (in the second row) and abnormal appearance (the eyes in the first row). On the contrary, SDM and our method localize the eye corners and mouth corners very accurately despite facial expres-

sion and occlusion being presented. Our method also localizes the more difficult landmarks along the face contour very accurately.

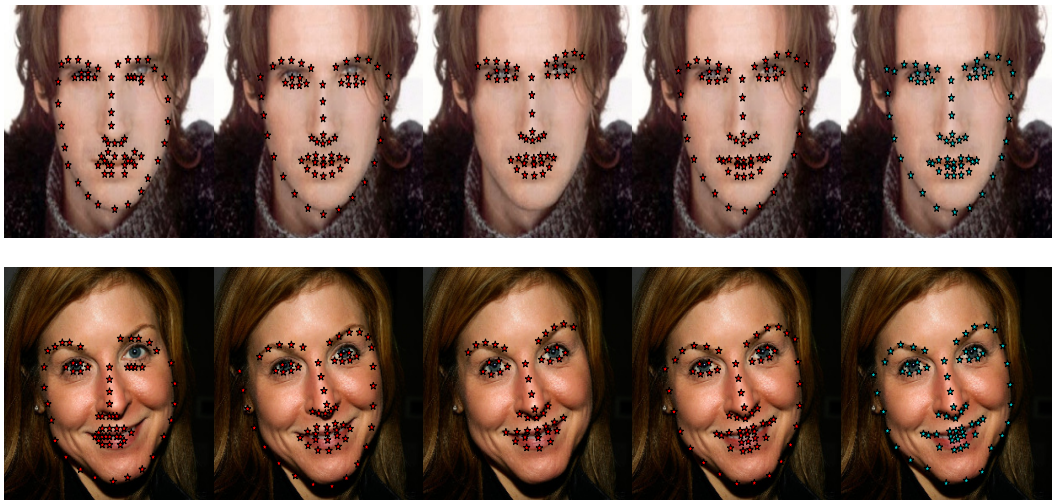


Fig. 3.6 Example results of different methods on LFPW (the first row) and on HELEN (the second row).

3.3 Random Subspace based Supervised Descent Method

In previous section, we have presented a cascaded framework that uses a regression forest as the primitive regressor. It can be regarded as a non-linear regressor built on top of raw pixel intensity features. In this section, we investigate a linear regressor based on hand crafted features, inspired by the Supervised Descent Method (SDM) (Xiong and De la Torre, 2013). We present our Random Subspace Supervised Descent Method (RSSDM) that maintains the high accuracy on the training data and improves the generalization accuracy of SDM. Instead of using all the features for descent learning at each iteration, we randomly select sub-sets of the features and learn an ensemble of descent maps in the subspaces, one in each sub-set. Then, we average the ensemble of descents to calculate the update of the iteration. We test the proposed methods on two representative problems, namely, 3D pose estimation and face alignment and show that RSSDM consistently outperforms SDM in both tasks in terms of accuracy. RSSDM also holds several useful generalization properties: e.g. it is more effective when the number of training samples is small and less sensitive to the changes of the strength of the regularization.

3.3.1 Problem definition

Many problems in Artificial Intelligence, such as image alignment in computer vision, can be posed as non-linear optimization problems. One of the most successful methods is Newton's gradient descent method. However, when it is applied to computer vision problems, there are many drawbacks of this second order optimization scheme. For example, some popular features like the HOG (Dalal and Triggs, 2005) and the SIFT (Lowe, 1999) are not twice differentiable. Moreover, computation of the Jacobians and Hessians is very expensive. To tackle such issues, recently a Supervised Descent Method (Xiong and De la Torre, 2013) (**SDM**) is proposed. Similar to Newton's method, given an initial estimate of the state of an object $S_0 \in \Re^{p\times 1}$ (e.g. this can be a p dimensional 3D pose vector of an object, or a 2D shape vector representing the locations of facial landmarks in an image), SDM creates a sequence of descent maps $\mathbf{R}_0, \cdots, \mathbf{R}_k, \cdots$. Each update step is represented as:

$$S_{k+1} = S_k - \mathbf{R}_k(\mathrm{h}(S_k) - \mathrm{h}(S_*)) \tag{3.11}$$

where $\mathrm{h} : \Re^n \rightarrow \Re^m$, is a transformation that varies according to different applications. It can be regarded as a generalized feature extraction term. For instance, in face alignment case, $\mathrm{h}(S)$ represents the HOG values computed in the local patches extracted from the landmarks with shape S. In 3D pose estimation case, $\mathrm{h}(S)$ is the image projection of the 3D

model points. S_* represents an optimal solution. In this way, the learned sequence $\{\mathbf{R}_k\}$ moves the initial shape vector S_0 towards the optimal solution S_*. The key contribution of the SDM is the supervised learning of $\{\mathbf{R}_k\}$, that is based on a large number of training samples generated by a Monte-Carlo sampling methodology. In the proposed method, each update is given by a linear regressor, e.g. $\mathbf{R}_k$ are the parameters of linear functions of the features. Thus in (Asthana et al., 2014), this method is also called a *Sequential Cascade of Linear Regression*. The SDM has shown very good performances in several important computer vision problem such as 3D pose estimation and template tracking.

However, when developing the SDM model in practice, two main problems arise:

- In order to learn an optimal $\mathbf{R}_k$, at least m training samples are usually required, with m the dimensionality of the feature space. Otherwise, the system is under-determined. m is usually very big, for example, in the case of face alignment (Xiong and De la Torre, 2013) using HOG feature, $m = 66 \times 128$, with 66 the number of facial landmarks and 128 the length of a HOG feature associated with each landmark. Moreover, the closed-form solution of such equations requires the inversion of matrix of size $m \times m$, which is also computational expensive.

- The linear function, which maps very high dimensional feature space to very low dimension $\Re^m \rightarrow \Re^1$, is very likely to over-fit the data during the training time. Regularization is required therefore a free parameter needs to be tuned empirically. However, when both the number of samples and the feature space are big, a single linear regression struggles to avoid over-fitting a set of training data while maintaining good performance.

In this section, we propose a Random Subspace SDM (**RSSDM**) to overcome the drawbacks mentioned above and improve the generality, inspired by the *Stochastic Discrimination* theory (Kleinberg, 1990). **Random Subspace** was proposed in (Ho, 1995, 1998) for constructing trees in random forests, where significant improvements in accuracy were obtained. It is based on the theory of *stochastic discrimination* (SD) (Kleinberg, 1990) for a classification problem. SD is a general methodology for constructing classifiers appropriate for pattern recognition. It is based on combining an arbitrary numbers of very weak components, which are usually generated by some pseudo-random processes. It has the property that the very complex and accurate classifiers produced in this way retain the ability of their weak components, to generalize new data . For a given feature space of m dimensions, there are 2^m subspaces can be generated. Thus it is intractable to try all possible subspaces. The SD theory also shows that very high accuracy can be achieved far before all the possible weak

learner are used (Ho, 1998; Kleinberg et al., 1996). In this work we borrow the random subspace idea for the regression problem and we believe there is similar theoretic support.

At each iteration, instead of learning one linear regression, we learn several of them, each of which is based on randomly selecting a small number of dimensions from the feature space, e.g. a Random Subspace. Then, we use such an ensemble of linear regressors to represent the descent map. We test the proposed method in two representative application cases of the SDM, i.e., face alignment and 3D pose estimation to demonstrate the benefits of our approach. More specifically, our method (RSSDM): 1) can naturally handle the under-determined issue by transforming a full feature space into subspaces and shows significantly better performance when training samples are limited ; 2) shows great advantages in dealing with over-fitting and is more robust to regularization parameter changes; 3) can achieve monotonic increase in generalization accuracy w.r.t. the SDM and obtain performance superior or close to other recent methods.

3.3.2 Random subspace SDM

In the section, we first present the Random Subspace SDM for face alignment and then for 3D pose estimation. The main difference of those two applications is that, for face alignment, $x = \mathrm{h}(S_*)$ (i.e., the HOG features extracted from the optimal locations of facial landmarks) is unknown while for 3D pose estimation $x = \mathrm{h}(S_*)$ (i.e., the image projection under the optimal 3D pose) is known.

Random subspace SDM for face alignment

Similar to the setting of other face alignment models, at training time, a set of N images $\mathcal{I} = \{I_i\}_{i=1}^N$ are available, along with their ground truth locations of facial landmarks $\mathcal{S} = \{S_*^i\}$. Thus $S \in \Re^{2p\times 1}$, with p the number of facial landmarks. In what follows we refer to S as the *shape* of a face. Similar to most of the face alignment models, our method also assumes that the face detection is available both in the training and in test images. We represent the face bounding box from the face detector as $b^i = (b_c^i, b_w^i, b_h^i)$, with $b_c^i \in \Re^2$ the face center, b_w^i the width and b_h^i the height. Then the location of the j-th landmark vector $S^{i,j}$, containing the x and y coordinates, can be translated by the box center and scaled by the box size, which we will refer to as normalized by b^i:

$$\mathcal{N}(S^{i,j}; b^i) = \begin{pmatrix} \frac{1}{b_w^i} & 0 \\ 0 & \frac{1}{b_h^i} \end{pmatrix} (S^{i,j} - b_c^i) \tag{3.12}$$

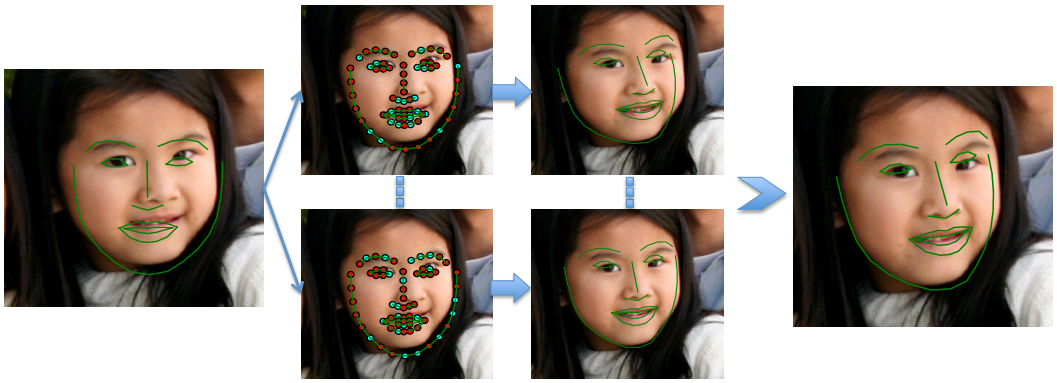


Fig. 3.7 RSSDM for face alignment. The image on the left shows the current pose. Then several subspaces are randomly generated, of which the cyan landmarks are selected and the red are not selected. Each regressor generates an update of the shape vector. The results are averaged as the final prediction of this iteration, as shown in the image on the right.

Since the face box provides the scale information, the image is transformed as to ensure the face box is at a canonical size (width and height), which is denoted by $(\bar{b}_w, \bar{b}_h)$. The scale factors are (s_w^i, s_h^i) with $s_w^i = \frac{\bar{b}_w}{b_w^i}$, $s_h^i = \frac{\bar{b}_h}{b_h^i}$. The initial shape estimate is given by centering the mean face at the canonical face box, that is denoted by S_0. In the rest of this Chapter, we assume the shape vectors and the images are transformed by the face boxes. We also generated 10 samples by perturbing the face box for each training image using Monte Carlo methodology as described in (Xiong and De la Torre, 2013). This augments the training samples by a factor of 10. We will treat each of them as a unique sample. Then for the i-th sample, the desired update (error vector) is $\Delta S_0^i = S_*^i - S_0^i$. HOG features around each landmark under the current shape are extracted $\widetilde{\phi}_0^i = h(I^i, x_0^i)$. Since S_* is not available for this problem we added a bias term to the feature vector for linear regression so that the feature vector becomes $\phi_0^i = [(\widetilde{\phi}_0^i)^T, 1]^T$. Thus we seek for $\mathbf{R}_0$ that minimizes:

$$\arg\min_{\mathbf{R}_0} \sum_i ||\Delta S_0^i - \mathbf{R}_0 \phi_0^i||^2 \tag{3.13}$$

The above least squares problem can be solved in closed-form given sufficient samples (equations). Then by applying the learned regressor $\mathbf{R}_{k-1}$, we can update the current shape S_k^i by adding the update. The new optimal update becomes $\Delta \mathrm{x}_k^i = S_*^i - S_k^i$ and the new feature vector is ϕ_k^i. A new regressor $\mathbf{R}_k$ can be learned by minimizing:

$$\arg\min_{\mathbf{R}_k} \sum_i ||\Delta S_k^i - \mathbf{R}_k \phi_k^i||^2. \tag{3.14}$$

As discussed before, the feature dimension is usually very high, thus it is easy to over-fit the model during optimization. Thus in practice, a regularization term should be added to prevent over-fitting and the optimization becomes:

$$\arg\min_{\mathbf{R}_k} \sum_i ||\Delta S_k^i - \mathbf{R}_k \phi_k^i||^2 + \lambda ||\mathbf{R}_k||_F^2. \tag{3.15}$$

This formulation requires tuning the λ therefore cross validation is usually applied to search for the optimal λ. However when the size of training samples are large, which is to guarantee closed-form solution, selecting a proper λ is intractable. Encouraged by the success of Random Subspace in tree construction (Ho, 1998), which also faces the over-fitting issue, we adapt it for the SDM. More specifically, instead of using the whole feature space, we select several random subspaces and train an ensemble of regressors in subspaces. For this face alignment case, we still keep the feature structure extracted from one landmark location. As shown in Fig. 3.7, from the set of landmarks $J = \{j\}_{j=1}^p$, we select several subsets, $\{J_t\}_{t=1}^T$, with $J_t \subset J$. We denote the features exacted from the landmarks in the t-th subset as $\phi_k^{i,t}$, $\phi_k^{i,t} \subset \phi_k^i$. We then train T regressors, one on each subset, using the corresponding features. We then optimize the following function:

$$\arg\min_{\mathbf{R}_k^t} \sum_i ||\Delta S_k^i - \mathbf{R}_k^t \phi_k^{i,t}||^2 + \lambda^t ||\mathbf{R}_k^t||_F^2. \tag{3.16}$$

for each of $\mathbf{R}_k^t, t = 1, ..., T$, regressors. We then simply average the outputs of such an ensemble of regressors to update the current shape. That is

$$S_{k+1}^i = S_k^i - \sum_{t=1}^T \mathbf{R}_k^t \phi_k^{i,t}. \tag{3.17}$$

A recursive procedure similar to the SDM is applied in the cascade framework when the shape of each sample is updated until the final iteration is applied. During testing time, since we have normalized the image using Eq. 3.12, we apply the inverse of of the normalization function to transform the final shape vector and obtain the alignment result. Assuming that the shape estimation after applying the final iteration is S_K^i, then the final shape estimation is:

$$\hat{S}^i = \mathcal{N}^{-1}(S_K^i; b^i). \tag{3.18}$$

Random subspace SDM for 3D pose estimation

In this section we present how we apply the random subspace SDM to another computer vision problem, 3D pose estimation. This problem can be described as follows. Given the 3D model of an object represented as 3D points $\mathrm{M} \in \Re^{3\times p}$, its image projection $\mathrm{U} \in \Re^{2\times p}$ and the intrinsic camera parameters $\mathrm{K} \in \Re^{3\times 3}$, the goal is to estimate the 3D object pose, consisting of a rotation vector ($\theta \in \Re^{3\times 1}$) and a translation vector $tr \in \Re^{3\times 1}$. To be consistent, we denote the pose vector by $S = [\theta; tr]$. Then the objective function becomes $||\mathrm{h}(S,\mathrm{M}) - U||_F$, with a known K. Given a set of poses $\{\mathrm{x}_*^i\}$ and the image projections U^i, the SDM optimization is defined as:

$$\arg\min_{\mathbf{R}_k} \sum_i ||S_*^i - S_k^i + \mathbf{R}_k(\mathrm{h}(S_k^i,\mathrm{M}) - \mathrm{U}^i)||_2^2. \tag{3.19}$$

Similar to the RSSDM for face alignment, we propose to use an ensemble of regressors in subspaces at each iteration. We denote by $\phi_k^i = \mathrm{h}(S_k^i,\mathrm{M})$ the features extracted based on the current pose S_k^i, and $\phi_k^{i,t}$ the feature in subspace t. The corresponding image projection is $\mathrm{U}^{i,t}$. Similar to Eq. 3.16, the optimization of the regressor in subspace t is as follows,

$$\arg\min_{\mathbf{R}_k^t} \sum_i ||S_*^i - S_k^i + \mathbf{R}_k^t(\phi_k^{i,t} - \mathrm{U}^{i,t})||_2^2 + \lambda^t ||\mathbf{R}_k^t||_F^2. \tag{3.20}$$

The update of the pose is calculated in a way similar to Eq. 3.17. At testing time, with a sequence of descent maps, the RSSDM always starts at the mean pose S_0, similar to the SDM method, and converges to the optimal solutions.

3.3.3 Evaluation

In order to evaluate the proposed RSSDM method, we carry out the experiments on face alignment and 3D pose estimation separately.

Face alignment

We first evaluate the RSSDM for face alignment, where the SDM has demonstrated state of the art result. The experiment is carried out on the most challenging datasets collected in the wild, namely the 300W. For method evaluation, we only use the training images in HELEN to train the baseline SDM model and different variants of our method. For comparison to the state of the art, we set up the experiments following the way of a recent method (Ren et al., 2014): the training set is split into two parts. More specifically, the training part consists of AFW, the training images of LFPW and the training images of HELEN, 3148 samples

in total. The XM2VTS set is not used in our method as it is taken from very constrained environment and is not publicly available. The testing set consists of the test images of LFPW, the test images of HELEN and the images in the iBug set, 689 samples in total. The test set is further partitioned into a Easy-set (LFPW and HELEN test images) and a Challenging-set (iBug images).

We implement the baseline SDM following the description in (Xiong and De la Torre, 2014) and by communication with their authors for some details. Ridge Regression is used for linear regression model learning at each iteration. Our RSSDM is trained in a similar way using the features in the randomly selected subspace. The landmark-wise localisation error is normalized by the face size if not explicitly stated otherwise, as suggested by (Sagonas et al., 2013c).

RSSDM parameters In the RSSDM framework, there are many interesting questions. For instance, how many of the subspaces are appropriate for a given application? What is the optimal size of the subspace? It is difficult to address them in a thorough theoretical way though (Kleinberg et al., 1996) has discussed some of them. In our work, we do a grid search for these parameters. More specifically, we set the number of subspaces in the range of $N_{SP} = [2:2:10]$ and the subspace feature dimensionality in the range of $D_{SP} = \frac{D}{[2:1:6]}$, where D is the dimensionality of the original feature space. Each combination of them is evaluated separately and we report their results in Fig. 3.8. When the number of subspaces is very low, decreasing the subspace dimension (using less features) will lead to larger error. When the number of subspaces is at a moderate number (6 or 8), the optimal subspace dimension lies in the middle. We select the second best combination of ($N_{SP} = 6$ and $D_{SP} = \frac{1}{3}$) in our following experiments as it has similar run-time cost as the original SDM while keeping good performance.

RSSDM vs. SDM In order to evaluate our Random Subspace strategy, we conduct the experiments and compare to SDM on HELEN dataset from 300W. We use the annotation provided by 300W in order to make the further comparison easier. It consists of 2000 images for training and 330 for testing, exhibiting large variations of head poses, illumination conditions, facial expressions and occlusion patterns. We use their default splitting of the data set and train the SDM and our RSSDM on the training images. For a fair comparison, we use the same experiment setting for model building, i.e., the same training images and the same distribution of 10 permutations for each training sample. In order to train an optimal model for both SDM and RSSDM, at each iteration we search for the optimal penalty parameter in a big space by 10-fold cross validation. In order to keep the proposed method

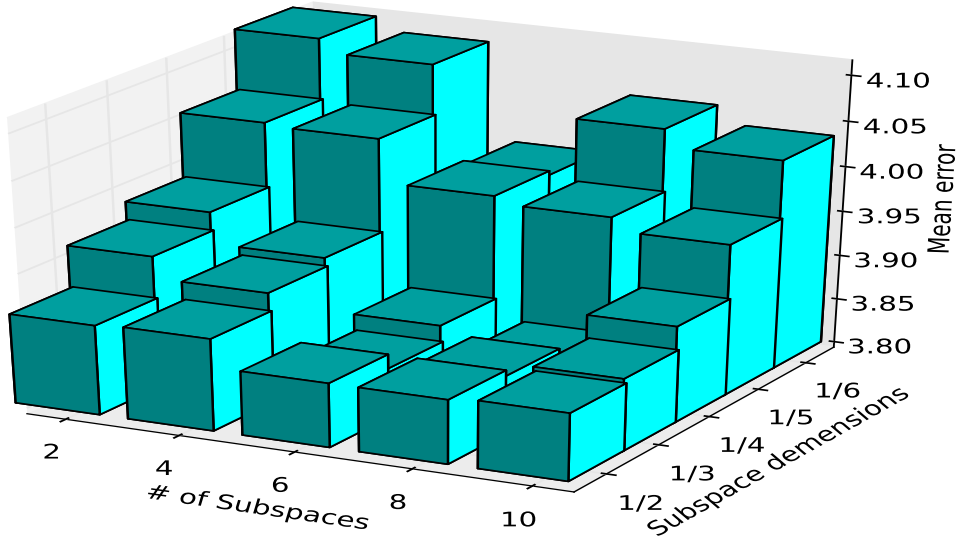


Fig. 3.8 RSSDM performance with various number of subspaces and subspace dimensions.

as simple as the original SDM, we set the dimensionality of the subspace to a fixed number, that is $\frac{1}{3}$ of the original feature space and 6 random subspaces. In the training process, we terminate the cascade when the error on the training set is lower than a threshold. In this way, we get very close training error for SDM and RSSDM. Then we test the model on the test images on both the Easy-set (images from LFPW and HELEN) and Challenging-set (iBug images). As the results shown in Fig. 3.9, RSSDM consistently performs better than SDM on both the Easy set and the Challenging set. Since the performance on the Easy set is near saturation, with the detection rate close to 100% at the error rate of 0.15, the improvement of RSSDM over SDM is small. The improvement on the Challenging set is larger, with 3% improvement at error rate of 0.1. Though the overall improvement is not huge, as we will show in the following, the proposed RSSDM scheme has benefits in certain circumstances, while still keeping monotonically increasing performance in accuracy w.r.t. SDM.

Sensitivity to number of Monte-Carlo permutations In this section, we compare the performance of RSSDM and SDM when the permutation number changes. As stated in (Xiong and De la Torre, 2014), the generic DM only exists within a local neighbourhood of the optimal parameters. Therefore in the training process, the number of Monte-Carlo permutations affect the results significantly. In this section, we evaluate the sensitivity of our RSSDM and SDM to the Monte-Carlo number. We set the system parameters including the regularization parameters and the number of iterations of both methods to the optimal

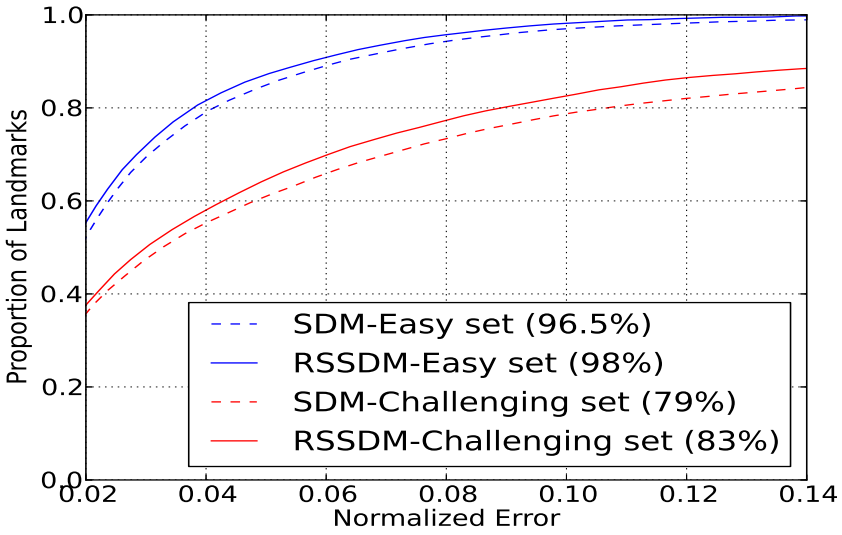


Fig. 3.9 RSSDM vs. SDM. The models are trained on training images in the HELEN dataset. The percentages in the legend show the proportion at the error level of 0.1.

ones learned from the above section. Then we decrease the permutation number from 9 to 1 with step size 2 and calculate the performance on the Easy and Challenging test sets respectively. The result is shown in Fig. 3.10. As expected, the error of both the SDM and RSSDM increases while the number of Monte-Carlo permutations decreases. However, the impact on RSSDM is less. On the easy set, the mean error increases from 2.74% to 2.85% for RSSDM while that of SDM increases from 2.83% to 3.26%. On the Challenging set, the mean error of RSSDM increases from 8.55% to 9.03% while that of SDM increases from 8.89% to 10.22%. Based on this observation, we can make the conclusion that, the proposed RSSDM is less sensitive to Monte-Carlo permutation reduction. Another conclusion we can draw from this experiment is that, RSSDM is able to obtain better performance when the training samples are limited. RSSDM with 3 Monte-Carlo permutations can achieve similar performance to SDM with 9 Monte-Carlo permutations. This is a very useful property under the circumstance when it is intractable to generate a large number of Monte-Carlo samples.

Sensitivity to λ In this section, we measure the sensitivity of RSSDM and SDM to the regularization parameter λ. In the previous discussion, we have obtained the optimal λ at each iteration. Assuming that the optimized λ is λ^*, we retrain the models using λ with the following values $[0.1\lambda^*, 0.5\lambda^*, \lambda^*, 5\lambda^*, 10\lambda^*]$ and record their results. As can be seen in Fig. 3.11, when the regularization parameter shift from the optimal one, the error for both RSSDM and SDM increases. However, RSSDM shows better performances in terms

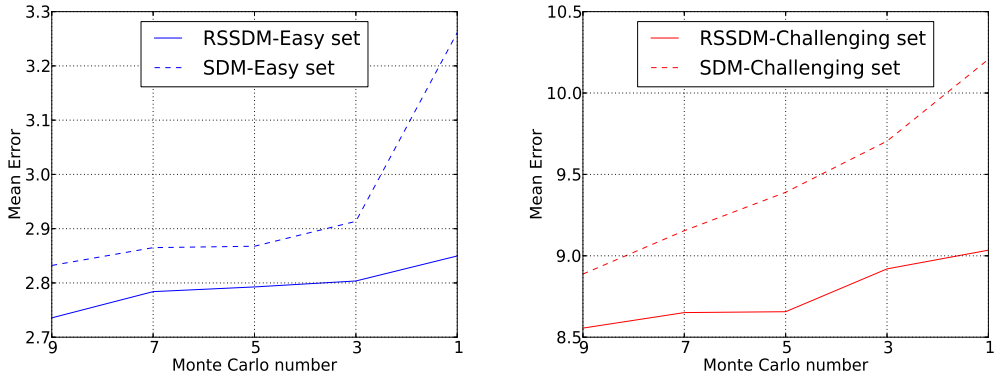


Fig. 3.10 RSSDM vs. SDM results with various Monte-Carlo numbers for model building. The models are trained on training images in the HELEN dataset. The figure on the left shows the results on Easy test set and the figure on the right shows the results on Challenging test set.

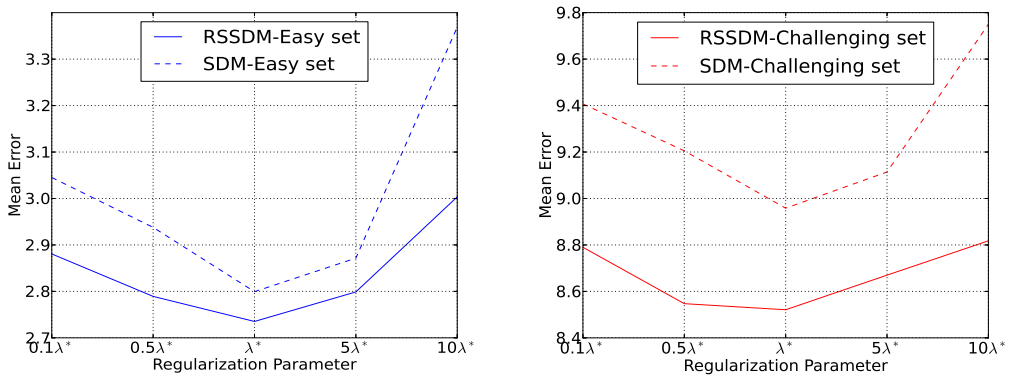


Fig. 3.11 RSSDM vs. SDM results with various regularization parameters, where λ^* is the optimized regularization parameter.

of robustness to such changes. For instance, on the easy set, when λ changes from λ^* to 10 times larger, the mean error of SDM increases nearly 0.6 while that of RSSDM increases only 0.25. On the Challenging set, the error increase of SDM is 0.8 while that of RSSDM is only 0.3. This can be explained by the ensemble strategy of the RSSDM method, of which in each iteration, the update is an average of the outputs from several weak regressors.

Comparison to state of the art Face alignment is a very active research topic in the field of computer vision and several recent methods have reported very good results. In this section we compare the performance of our proposed RSSDM to them. We conduct the experiments in two scenarios. First, we follow the setting of a very recent paper (Ren et al., 2014), which is based on 300W dataset. We compare the RSSDM with the most competitive

methods including the baseline Supervised Descent Method (SDM) (Xiong and De la Torre, 2013), the Explicit Shape Regression (ESR) method (Cao et al., 2012), the Robust Cascaded Pose Regression (RCPR) in (Burgos-Artizzu et al., 2013), the Local Binary Feature (LBF) method (Ren et al., 2014) and its fast version fast-LBF. In this scenario, the localisation error of all 68 facial landmarks are recorded. In order to be consistent to (Ren et al., 2014), the error is normalized by the inter-ocular distance instead of the face size in this experiment. The results of SDM, ESR, LBF and LBF-fast are quoted from (Ren et al., 2014). We re-train the RCPR model on the same experimental setting but we do not consider the occlusion status since no occlusion annotation is provided. The results are shown in Table 3.4. As can be seen, the proposed RSSDM outperforms SDM and most other current methods and has comparable performance to LBF. However, we note that LBF has used very different learned features and its model, consisting of thousands of trees, is much more complex.

In the second scenario, we compare the RSSDM performance with state of the art publicly available models that include the SDM (SDM-A) (Xiong and De la Torre, 2013), the Incremental Face Alignment (IFA) model in (Asthana et al., 2014) and our implementation of SDM (SDM-B). In this scenario, only the 49 inner facial landmarks are localized and normalized by the inter-ocular distance, as the the publicly available models do. The results are shown in Table 3.5. Our implementation of SDM performs slightly better than the publicly available model. The IFA is a variant of SDM that can be trained in a parallel way and also trained on the 300W dataset using HOG features. The proposed RSSDM outperforms the two versions of SDM as well as the IFA.

Table 3.4 300-W dataset (68 landmarks).

Method	Full-set	Easy-set	Challenging-set
ESR	7.58	5.28	17.00
RCPR	7.54	5.67	15.50
LBF fast	7.37	5.38	15.50
LBF	6.32	4.95	11.98
SDM	7.52	5.60	15.40
RSSDM	6.58ł	5.11	12.61

Table 3.5 300-W dataset (49 landmarks).

Method	Full-set	Easy-set	Challenging-set
IFA	8.30	5.48	19.88
SDM-A	7.06	5.56	13.22
SDM-B	6.86	5.45	12.66
RSSDM	6.17	4.95	11.20

3D pose estimation

In this section we evaluate the performance of RSSDM on another computer vision problem, 3D pose estimation. As we discussed before, our method is proposed for the situation that the feature space is much bigger than the output space. Thus we use a human body 3D pose estimation [1] in our experiment to demonstrate the performance. As shown in Fig. 3.12, the 3D body consists of 996 3D key points. Then its image projection contains 996×2 dimensions of features for descent map learning. We follow the same experimental setting as (Xiong and De la Torre, 2014). More specificity, the virtual camera is at the origin of the coordinate system and the intrinsic parameters are: focal length $f_x = f_y = 1000$ pixels, principle point $[u_0, v_0] = [500, 500]$. The object is placed at $[0, 0, 2000]$, and perturbed with different 3D poses. Three rotation angles are uniformly sampled from 30^o to 30^o with increments of 10^o in training and 7^o in testing. Three translation values are uniformly sampled from -400mm to 400mm with increments of 200mm in training and 170mm in testing. For each combination we get one training sample. We also add white noise ($\sigma^2 = 4$) on the projected points and normalize the projection by the focal length and the principle point of the camera. We also do a grid search for both SDM and RSSDM for the optimal parameters by cross validation on the training set. The result is shown in Table 3.6. As a re-implementation, our result of SDM is slightly different from (Xiong and De la Torre, 2014). As can be seen in the figure, both SDM and RSSDM outperform the POSIT algorithm with a large margin. The RSSDM further improves the accuracy over SDM, which validates the efficacy of our proposed method in 3D pose estimation application.

Table 3.6 Rotation (in degree) and translation (in mm) errors of 3D body pose estimation.

Method	θ_x	θ_y	θ_z	tr_x	tr_y	tr_z
POSIT	0.6±0.6	6.3±5.3	2.1±1.6	22.3±14.8	14.9±11.2	41.1±38.0
SDM	0.07±0.05	0.25±0.15	0.2±0.11	3.7±3.0	4.1±3.6	6.5±5.3
RSSDM	0.06±0.04	0.22±0.13	0.15±0.09	3.4± 3.1	3.7±3.2	5.2±4.3

3.4 Summary

In this chapter, we propose three holistic methods for face alignment. First, we show the efficiency of using regression forests as the primitive regressor in the Cascaded Pose Regression framework. We propose an intelligent initialization scheme that is able to select a few reliable pose estimations in a few stages in the cascade and aggregate them to the remaining

[1] http://www.robots.ox.ac.uk/~wmayol/3D/nancymatlab.html

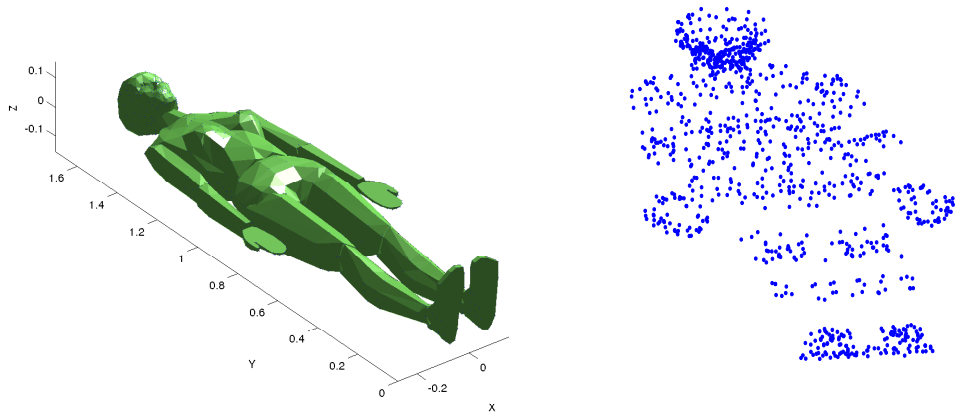


Fig. 3.12 3D human body and its image projection of the 3D points under a certain pose.

cascade to calculate the final pose. Furthermore, we have shown that through using different features, there is a slight improvement of performance at the cost of feature computation. Also, as the cascaded method is sensitive to initialization that is calculated from face detection, our method to some degree is capable of decreasing the risk by using the intelligent initialization scheme. Second, we propose a simple yet effective Random Subspace SDM (RSSDM). We compare RSSDM to SDM on two representative problems, namely, Face Alignment and 3D pose estimation and obtain better performance in estimation accuracy. It also holds several other interesting properties, i.e., RSSDM is more effective than SDM when the Monte-Carlo number is small and less sensitive to the regularization term, which, we believe are important in designing a real system.

By using the proposed methods in the section, we have achieved performance superior or on par with the state of the art. It also demonstrates better performance than the local based on methods we proposed earlier in terms of both accuracy and efficiency. However, they are intrinsically more sensitive to initialization than the local-based method. Thus the initialization of cascaded method is an interesting problem that needs to be further investigated in future work. Furthermore, it still shows failures, usually also in a holistic way, when heavy occlusion is presented.

Chapter 4

Robust Face Alignment Under Occlusion

In previous chapters, we have presented both local-based and holistic based methods for face alignment in the wild. We have dealt with a wild range of challenges like head pose changes, small amounts of occlusion and multi-modality. Most of the proposed methods show very good performance on images with limited occlusion. However, their performance deteriorates significantly when applied on images with heavy occlusion since their models cannot handle missing features.

Despite the fact that face images in real world are frequently occluded by objects like sunglasses, hair, hands, scarf and other unpredictable items, only a few works have explicitly addressed the occlusion issue (Ghiasi and Fowlkes, 2014; Roh et al., 2011; Yang et al., 2011; Yu et al., 2013b). Those works focus on synthesized data or consider a very limited number of occlusion patterns (sunglasses, scarf and hands) and assume that only a small portion of the face image is occluded. However, in real scenarios, the occlusion patterns can be very diverse. Burgos-Artizzu et al. (Burgos-Artizzu et al., 2013) carried out a pilot work in face alignment under occlusion. It proposed an occlusion-centered approach that leveraged occlusion information to improve the robustness of the CPR method, however, it cannot deal with the large diversity of the occlusion patterns. It also only provides an occlusion label for each landmark, however, the occlusion often covers a region in practice.

In this chapter, we present two methods specifically for face alignment under heavy occlusion. The first one, as presented in Section 4.1, models the occlusion in an supervised way thus it is based on a subset set of images annotated with face masks. It forms a structured semi-supervised joint classification-regression forest. The second one, as presented in Section 4.2, models the occlusion in an unsupervised way and exploits the regression consistency in face regions returned from image segmentation.

4.1 Supervised Occlusion Modelling for Face Alignment

4.1.1 Problem definition

In this section, we address face mask reasoning and facial landmark localisation (face alignment) in an unified random Decision Forests (DF) (Criminisi et al., 2011b) framework. As we sated in Section 2.3, not all votes from the forest are valid and the invalid votes degrade the localisation accuracy. In our observation, these invalid votes are very likely from the occluded facial regions. Therefore, we model patch occlusion status explicitly, in a way similar to semantic image labelling (Dollár and Zitnick, 2013; Kontschieder et al., 2011, 2013), by encoding each pixel with a semantic label, face or non-face in our case. We propose a structured semi-supervised forest framework for face mask reasoning and landmarks localisation. In order to model the occlusion explicitly, we built a rich face image dataset with face mask annotation. The dataset was built as an extension of the recent datasets: Caltech Occluded Faces in the Wild (COFW), Labeled Face Parts in the Wild (LFPW) and Labeled Face in the Wild (LFW). We manually annotate a portion of images in these datasets with face masks. The face mask indicates whether or not each pixel belongs to the face. We propose a structured semi-supervised joint classification-regression forest with the following properties. First, semi-supervised, it uses training images from the above described augmented dataset, only a portion of which are with face masks. Second, it has a novel structured criterion for split function selection for the pixel labelling (face mask reasoning) problem. Third, joint classification-regression, it predicts face mask label for each pixel (classification) and the landmark locations (regression) at the same time, and more importantly it uses the face mask reasoning results to improve the accuracy of landmark localisation.

Our structured semi-supervised forest performs classification and regression on their corresponding domains in one estimator as we believe these two tasks are mutually dependent. We start with a brief introduction of the augmented training data in Section 4.1.2. Then, we show how we encode both the landmarks locations and structured face/non-face labels within the decision forests in Section 4.1.3. Finally, we describe the inference procedure in Section 4.1.4.

4.1.2 Training preparation

A forest is an ensemble of trees $\mathcal{T} = \{T_t\}$. Each tree T_t is built on a randomly selected subset of the training images. In our semi-supervised setting, we have a portion of images with face mask labeling and the rest without. We randomly extract a set of training data (patches) from the training images. We denote it by $\mathcal{D} = \{\mathcal{P}_l, \mathcal{P}_u\}$, where $\mathcal{P}_l$ represents

Fig. 4.1 The images on the left side of the two pairs show the results from the standard Random Forests for facial landmarks localisation (Dantone et al., 2012b), with failure cases under occlusion. The images on the right side of the two pairs show the results of our proposed method. It first explicitly predicts the face mask (the semi-transparent region), then use the face mask information to improve the localisation and to predict the occlusion status of the landmarks.

the patch extracted from training images with face mask label and $\mathcal{P}_u$ represents that from training images without face mask label. Without loss of generality we denote them in the same form $\mathcal{P} = (\mathcal{I}^{d\times d\times F}, \mathcal{V}^{2\times N}, \mathcal{M}^{d'\times d'})$, where $\mathcal{I}$ is the $d \times d$ sized image patch with F channels of features; $\mathcal{V}$ is a N $2D$ displacement vector from the patch centroid to each of the N facial landmarks; $\mathcal{M}$ consists of the $d' \times d'$ of class labels, i.e., $\mathcal{M} = \mathcal{Y}^{d'\times d'}$. Note that the size d' of label patch may differ from the size d of the image patch. For $\mathcal{P}_u$ where there is no face labels, $\mathcal{M}$ is a null matrix.

4.1.3 Structured decision forests

In this section, we demonstrate how to encode both the landmarks locations and structured face labels (face mask) in the learning procedure of decision forests. Of particular interest in this work is the case where $x \in \mathcal{X}$ represents input image patch and $y \in \mathcal{Y}$ encode the corresponding image annotation (in our case, $\mathcal{Y} = \mathcal{V} \times \mathcal{M}$, where $\mathcal{V}$ is the landmark offset vector and $\mathcal{M}$ is the face mask). Thus, we have two objectives: first, localisation of the landmarks and second, the structured labels of different classes (face or non-face). Similar to the hybrid forests (Tang et al., 2013), we use two separate types of split nodes that optimize different objective functions. The first type of node is for regression and the second type is for classification.

Specifically, for a given node i and the training set $\mathcal{D}_i \subset \mathcal{X} \times \mathcal{Y}$, the goal is to find the best split function $h(x, \theta_i)$ with parameters $\theta_i = (f, k_1, k_2, \tau)$ from a pool of randomly generated candidates, where f is the feature channel, k_i is the sub-region within patch and τ is the threshold,

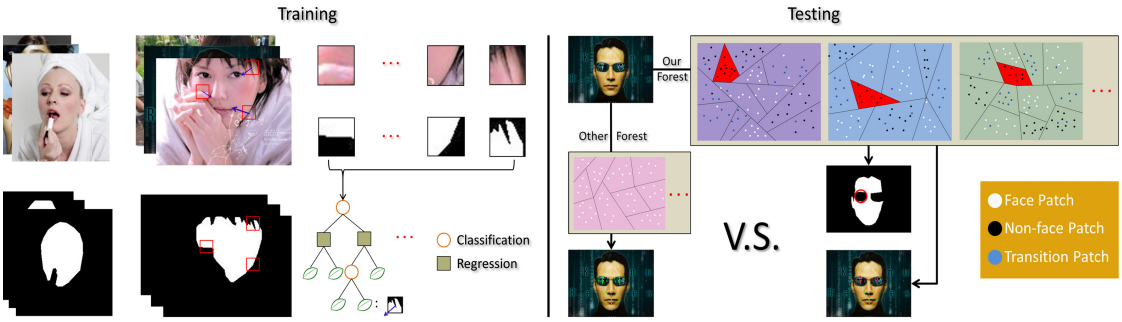


Fig. 4.2 The framework of proposed method. We use face images with annotation of facial landmarks and face masks for training. By randomly switching the information gain function at the internal nodes, the decision trees are optimized with respect to both the offsets to landmarks (regression) and to the local structured label configuration (classification). The forest model is able to predict the face mask and landmark locations jointly. We exploit the face mask prediction to further improve the landmark localisation.

$$h(x,\theta_i) = \begin{cases} 0 & \text{if} \quad x^f(k_1) < x^f(k_2) + \tau \\ 1 & \text{otherwise} \end{cases} . \tag{4.1}$$

that maximizes an objective function, in our case the information gain:

$$I(\mathcal{D}_i, \mathcal{D}_i^L, \mathcal{D}_i^R) = H(\mathcal{D}_i) - \sum_{j \in L,R} \frac{|\mathcal{D}_i^j|}{|\mathcal{D}_i|} H(\mathcal{D}_i^j). \tag{4.2}$$

where $H(\cdot)$ is the entropy function. The same procedure is applied recursively on the child nodes, $\mathcal{D}_i^L$ and $\mathcal{D}_i^R$, until a certain stopping criterion is met, for instance when a maximum depth is reached or the information gain or training data size fall below fixed thresholds.
For regression nodes, we need to adapt information gain calculation for continuous variables. In our case for $\mathcal{V}$, the aim is to cast precise votes concerning the landmarks location. Therefore, we follow the class-affiliation method proposed by (Dantone et al., 2012b) to measure the uncertainty which is defined as:

$$H_{\mathcal{V}}(\mathcal{D}_i) = -\sum_{n=1}^{N} \frac{\sum_{\mathcal{P} \in \mathcal{D}_i} (c_n|\mathcal{P})}{|\mathcal{D}_i|} log \left(\frac{\sum_{\mathcal{P} \in \mathcal{D}_i} p(c_n|\mathcal{P})}{|\mathcal{D}_i|} \right) \tag{4.3}$$

$$p(c_n|\mathcal{P}) \propto \exp\left(\frac{|v^n|}{\lambda} \right), \tag{4.4}$$

where $p(c_n|\mathcal{P})$ indicates the probability that the patch $\mathcal{P}$ is informative about the location of the landmark point n. The class affiliation assignment is based on $|v|$, the Euclidean distance between the patch and the landmark location. The variable λ is used to control the steepness of this function.

For classification nodes, we propose a structured way of calculating the entropy. A standard classification method can only deal with a single (atomic) label per input patch sample. It usually represents the patch center label with a finite set of discrete class labels $(y \in \mathcal{Z})$. Consequently, $H(\cdot)$ is defined as the Shannon entropy

$$H(\mathcal{D}_i) = -\sum_{y} p(y|x) log(p(y|x)) \tag{4.5}$$

where $p(\cdot)$ is the empirical class distribution estimated from the training set $\mathcal{D}_i$. However, the abandoning of structured labels and making the prediction independently will result in inconsistency in the output spaces. For our face/non-face labeling problem, the unstructured prediction often results in inconsistent face mask reasoning. So far as $y \in \mathcal{Y}^{d' \times d'}$ is concerned, we face two main challenges: 1) information gain over structured label space is not well defined. 2) structured labels are often of high dimension, complex and prohibitively expensive to score numerous split candidates.

Inspired by recent works (Glocker et al., 2012), we define a structured criterion for split function selection. We first discretize the structured labels by partitioning the label spaces, that is inspired by the structured edge detection work of (Dollár and Zitnick, 2013). We utilize a two-stage approach. First we map the structured space to an inter-median space $\mathcal{B}$, $\mathcal{Y} \rightarrow \mathcal{B}$. Then we map the space $\mathcal{B}$ to a discrete label space $\mathcal{Z}$, $\mathcal{B} \rightarrow \mathcal{Z}$. More specifically, $\mathcal{B} = \Pi(\mathcal{Y})$ is a long binary vector that encodes whether every pair of pixels in $\mathcal{Y}$ belong to the same or different labels, such that we can approximately estimate the dissimilarity of $\mathcal{Y}$ by computing the hamming distance in space $\mathcal{B}$. Considering $\mathcal{B}$ may be high dimensional ($C_2^{d' \times d'}$ for a patch with $d' \times d'$ structured labels), dimensionality reduction is required for efficient computation. We first use a distinct and reduced mapping $\Pi_{\delta_i} : \mathcal{Y} \rightarrow \mathcal{B}$. Instead of using all pairs, we randomly generate m dimensions of $\mathcal{B}$, which is parametrized by δ_i and applied to the training set $\mathcal{D}_i$ at each node i. This not only contributes to fast computation but also introduces randomness into the learning process at the node level. After that, Principal Component Analysis (PCA) (Jolliffe, 2005) is applied to further project the reduced $\mathcal{B}$ to T dimensions.

Finally we map the entry in space $\mathcal{B}$ to a label in space $\mathcal{Z} = \{1, ..., k\}$, such that labels with similar $b \in \mathcal{B}$ are assigned to the same discrete labels z. We quantize b based on the top $log_2(k)$ PCA dimensions, assigning b a discrete label z according to the orthant

(generalization of quadrant) into which b falls. To this end, mapping the structured label to space $\mathcal{Z}$ allows us to use the standard information gain criterion based on Shannon entropy as defined in Eq. (4.5). In practice, we use Π_δ with dimension $m = 256$ and the discrete labels with $k = 2$. In fact, even an approximate distance measure for $\mathcal{Y}$ like this suffice to train effective decision forests classifiers (Geurts et al., 2006). We note that, in our semi-supervised setting, there are both $\mathcal{P}_l$ and $\mathcal{P}_u$, thus at classification nodes, the information gain is only evaluated on the data with labelled face mask. The entire learning procedure will greatly benefit from the contribution of the ones with unlabelled mask at regression nodes.

Leaf Models. As in Hough forests (Gall and Lempitsky, 2013), we assign certain levels of depth in the tree a fixed type of evaluation objective. We thus introduce a steering parameter γ which indicates from first levels up to depth $|\gamma|$, only those regression nodes are evaluated, such that the visual feature variation due to displacements to the facial points is first removed at top levels. Then, starting with depth $|\gamma|$ of the trees, classification nodes and regression nodes are selected randomly. Therefore, image patches reach one leaf node tend to have similar offsets to the facial points and exhibit similar structured face/non-face labels.

At each leaf node, e.g. leaf node l, we calculate: (i) the relative offsets to each facial point $O_l^n = (\Delta_l^n, \omega_l^n)$, similar to (Fanelli et al., 2011), where Δ_l^n is the mean value and $\omega_l^n = \frac{1}{trace(\Sigma_l^n)}$ with Σ_l^n the covariance matrix of the offsets to the nth facial landmark; (ii) a structured label y_l of size $d' \times d'$ based on $\mathcal{D}_l$ ($\mathcal{D}_l \subset \mathcal{D}$), which is a subset of training data at leaf node l. More specifically, we select the y_l ($y_l \in \mathcal{Y}$) whose value in the inter-median space $b_l \in \mathcal{B}$ is the medoid, i.e. the b_l that minimizes the sum distance to all other b in $\mathcal{D}_l$. This is equivalent to $min_l \sum_m (b_{lm} - \bar{b}_m)^2$, where $\bar{b}$ is the mean vector of all b in $\mathcal{D}_l$. We denote by $f_t^{\mathcal{C}}(x)$ the classification output of tree t cast by x and by $f_t^{\mathcal{R}}(x)$ the regression output.

4.1.4 Face mask reasoning and landmark localisation

At testing time, image patches $x \in \mathcal{X}$ are densely extracted with a stride s and fed to the forest until they reach leaf nodes, where votes are cast for both the localisation of facial points and the patch face/non-face label prediction. As opposed to standard classification algorithms, our classifier $f_t^{\mathcal{C}}(x)$ cast a prediction for the center pixel, as well as its neighbouring pixels. Hence, a predicted face mask $\mathcal{M}_p$ is obtained for each test image in a similar way to (Dollár and Zitnick, 2013). Specifically, each pixel gets $d' \times d' \times T/s^2$ predictions, where T is the number of trees and s is the stride size. Then we merge the multiple predictions by a simple average fusion to get the final face mask prediction. Meanwhile, given the regression

outputs of the forest, we can accumulate the Hough score for facial landmark n as follows. Denote each image patch x_y by its location y, which ends in a set of leaf nodes $\mathcal{L}_{x_y}$ in the forest.

$$Score(\hat{y}^n) \propto \sum_{x_y} \sum_{l \in \mathcal{L}_{x_y}} \omega_l^n \exp\left(-\| \frac{\hat{y}_n - (y + \Delta_l^n)}{h^n} \|_2^2\right) \cdot \delta_1(f(\Delta_l^n) > \lambda_n) \cdot \delta_2(\mathcal{M}_p(y) > \tau') \quad (4.6)$$

where h^n is a learned per-point bandwidth. $f(\Delta)$ is the proximity metric defined in Eq. (4.4). $\delta_i(\cdot)$ is the Dirac delta function. $\delta_1(\cdot)$ only allows votes which fulfil the proximity test, using the proximity threshold λ^n.

The face mask term $\mathcal{M}_p(y)$ differentiates our method from the existing works as we believe that the patches from face region and non-face region contribute differently to the facial landmarks localisation. The Dirac delta function $\delta_2(\cdot)$ isolates the effect of votes from non-face region which most likely correspond to the occluders. We note that, by setting $\tau' = 0$, we allow forests to collect votes from the entire image domain, while higher τ' only allows patches from face regions with higher face confidence.

Additionally, the ratio r^n: the sum of votes associated with each facial point n before and after the $\delta_2(\cdot)$ is applied is traced in our work. This is because for heavily occluded facial points, only few valid votes remain after $\delta_2(\cdot)$ is applied, so that the proximity threshold λ_n should be reduced to allow longer distant patches to cast their votes. Such votes essentially introduce stronger facial shape constraint. Finally a mean-shift mode finding algorithm is applied on the Hough map for final facial landmark localisation.

4.1.5 Experiment

Implementation details

We evaluate the performance of our proposed framework for both landmark localisation and face mask labelling on our augmented face image datasets on the LFW, LFPW and COFW.

Due to the performance saturation and the lack of occlusion on the LFPW and LFW dataset (Burgos-Artizzu et al., 2013; Dantone et al., 2012b), we cannot fully exploit the benefits of our face mask prediction and landmark localisations. Therefore, we only report the results on the 'difficult' subsets of LFW and LFPW. We obtained the difficult subsets in a similar way as section 2.3, namely the face images are regarded as difficult if the average point localisation error detected by the CRF-D (Dantone et al., 2012b) method is greater than 0.1 inter-ocular distance. 237 face images were obtained in the **LFW_Test** and 96 face images in the **LFPW_Test**. The number of resulting images is small due to the fact that face images on these two datasets are relatively easy. Only a few of them either contain

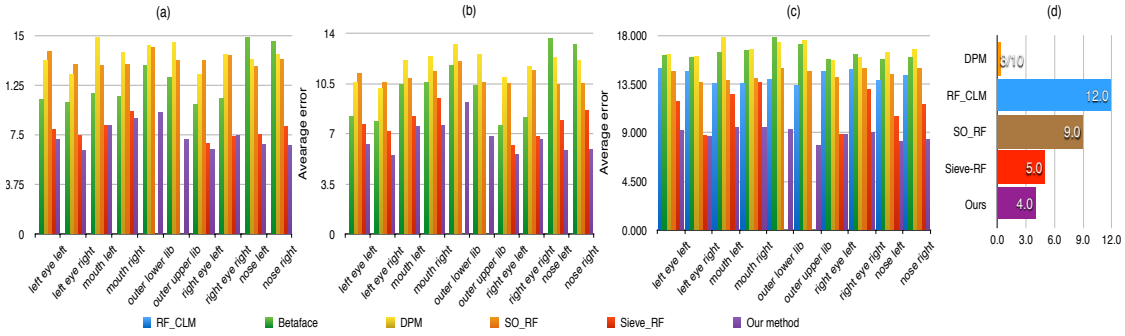


Fig. 4.3 Results on LFW_Test (a), LFPW_Test (b), and COFW (c), compared to (Betaface; Cootes et al., 2012; Zhu and Ramanan, 2012) and our previous methods from Section 2.2) and Section 2.3) . The error is measured as a fraction of the inter-ocular distance. LFW_Test and LFPW_Test only contain 'difficult' image. (d) shows the run-time performance in fps.

occlusion caused by hair or sunglasses, or present large shape variation. We also randomly select 300 images from COFW dataset as a test set. We note that, all images from the three test sets were annotated with masks.

We use all the remaining face images from the 3 datasets for model training, which consists of 6781 images and 1603 of them are with face mask labels. Each tree was built using 1200 images (nearly 600 of them with labelling mask) and 100 patches were extracted from each image with no labelling mask and 250 from the ones with labelling.

To build our forest model, we use similar experimental settings to (Dantone et al., 2012b) such as the face bounding box size, bandwidth parameter (4.6) and proximity threshold (4.6). Some other parameters are as follows: image patch size ($d = 24$), label patch size ($d' = 12$), 37 channels of image features (1 gray scale, 4 HOG-like features, 32 Gabor features), face confidence threshold ($\tau' = 0.78$). The macro forest parameters are: number of trees 10, steering parameter $\gamma = 7$, minimum number of samples 8, maximum depth 25.

Results for landmark localisation

We compare our method with the recent Decision Forest methods for facial feature detection. They are Regression Forests with Constrained Local Model (**RF_CLM**) (Cootes et al., 2012) and our work of Structured-Output Regression Forests (**SO_RF** in Section 2.2) and our Regression Forests Sieving (**Sieve_RF** in Section 2.3). We use the same experiment setting (image data, image feature and macro parameters of the forest) to re-train the Decision Forest models for SO_RF and RF_CLM. We also compare the representative DPM+tree structure method (Zhu and Ramanan, 2012) (**DPM**) and a commercial system (**Betaface**) (Betaface).

Fig. 4.3 shows the results of the 10 common facial landmarks on all three datasets.

Our proposed method achieved better performance than the other methods on 'difficult' images from LFW LFPW and the challenging COFW datasets, despite the fact that all the benchmark Decision Forest methods have used shape models, explicitly (SO_RF and RF_CLM) or implicitly (Sieve_RF) while our method only works as a local detector. On the COFW dataset, the performance of our method still has a gap to the performances of human, due to the heavy occlusion. Note that we have focused on comparing the Regression Forests voting method proposed in recent years, rather than on producing the best facial landmarks detector as we aim to validates the effectiveness of our proposed scheme, i.e., to select reliable patches from face regions based on face mask prediction. As our method is still a local detector, it can be naturally further combined with face shape models, for instance it can be combined with CLM in a way similar to (Cootes et al., 2012), in order to further boost the performance.

Our predicted face mask can intuitively reason about the occluded regions on a face image, rather than just checking the visibility (Burgos-Artizzu et al., 2013) of an individual pixel. We propose a more practical method for landmarks visibility detection. We calculated the occlusion ratio over a small region (within 0.2 inter-ocular distance) surrounding the estimated landmark location, and obtained a 80/57% precision/recall for landmark visibility prediction, which is much better than 80/40% reported in (Burgos-Artizzu et al., 2013).

Results for face mask reasoning

In this section, we evaluate face mask reasoning performance of our method on the COFW dataset. We compare to the methods that are used for general scene parsing: 1) the standard random forest which yield independent prediction (denoted by **Baseline RF**); 2) standard random forest + conditional random field post-processing (**BaselineRF+CRF**) (Kolmogorov, 2006); 3) three structured forest variants from (Kontschieder et al., 2011), namely: the **StructureRF+Simple Fusion**, the **FullRF+Simple Fusion** and the **FullRF+Optimized Selections**, all of which yield structured outputs. We followed the evaluation criteria as used in (Kontschieder et al., 2011). Specifically, two measurements are reported: *'Global'*, that refers to the percentage of all pixels that were correctly classified; *'Avg(face)'* that expresses the average recall over all classes (face and non-face).

We show the results in Table 4.1. First, we can clearly see a big margin between the standard RF and structured approaches, which enforce spatial consistency and yield plausible local configuration. Second, our structured approach outperforms the FullRF+Optimized selection and RF+CRF in terms of both *'Global'* and *'Avg(face)'*. The gain in performance validates the effectiveness of our proposed structured information gain criterion and the usefulness the joint classification and regression framework. Some results are shown in Fig 4.4.

Method	BaselineRF	BalineRF +CRF	StructureRF +Simple Fusion	FullRF +Simple Fusion	FullRF Opt. Sel.	Ours
Global	68.8	81.7	73.6	74.8	78.8	83.9
Avg (face)	71.2	86.6	74.2	75.1	81.7	88.6

Table 4.1 Face mask reasoning results on the COFW dataset, compared to the related methods.

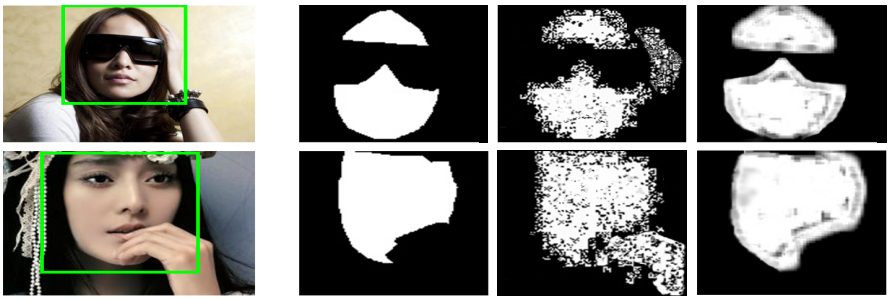


Fig. 4.4 Illustration of two face mask reasoning results on COFW: (from left to right) original image, ground truth, result of the standard RF and result of our proposed method.

4.2 Unsupervised Occlusion Modelling for Face Alignment

In previous section, we presented a forest framework for joint face alignment and facial mask prediction. It takes the advantage of the fact that a proportion of images are annotated with face mask labelling at the training stage. However, labelling the facial region at pixel level for training is very costly. Thus in this section, we present the framework for face alignment under occlusion that works in an unsupervised fashion, i.e., at the training stage, there is no additional annotation of the occlusion status. We present the concept of regional predictive power (RPP), that describes how useful each region from image segmentation is in the task of face alignment. The RPP is estimated by measuring the consistency of votes within each region.

4.2.1 Problem definition

Tackling the occlusion problem explicitly is difficult mainly due to two reasons. Firstly, compared to the intra-category shape variation of a face, the occluders[1] are much more diverse in appearance and shape. They can appear on the face in almost unpredictable arbitrary position with various sizes. Second, it is a chicken and egg problem since the occluders should not participate in the alignment but it is difficult to tell whether a landmark is occluded unless the correct alignment is known (Roh et al., 2011). Therefore, most of the existing works only consider the occlusion status of individual landmarks and treat the occlusion landmark as unstructured sources of noise. In addition, they require the annotation of occlusion during training, either annotated manually (Burgos-Artizzu et al., 2013) or synthesized artificially (Ghiasi and Fowlkes, 2014). These approaches show some success but have a series of drawbacks: (1) Treating the occlusion status of individual landmarks independently ignores a key aspect that the occluders are often other objects or surfaces and hence often appear in continuous regions instead of an isolated pixel. (2) The randomly synthesized occlusion patterns are not realistic enough to describe the occlusion diversity in real scenes. To collect face images with occlusions and to annotate their occlusion status is expensive, especially when a large number of such images are demanded for model training. (3) The occlusion detection at pixel level limits its practical application in face analysis since features are usually extracted from a region rather than an individual pixel.

The method presented in this work aims at dealing with face alignment under occlusion and overcome the above mentioned drawbacks. An overview of our method is shown in Fig. 4.5. Given a face image, the method starts by detecting the face and employing an

[1] In this thesis the objects that occlude the face are called occluders and the visible face region is called face mask.

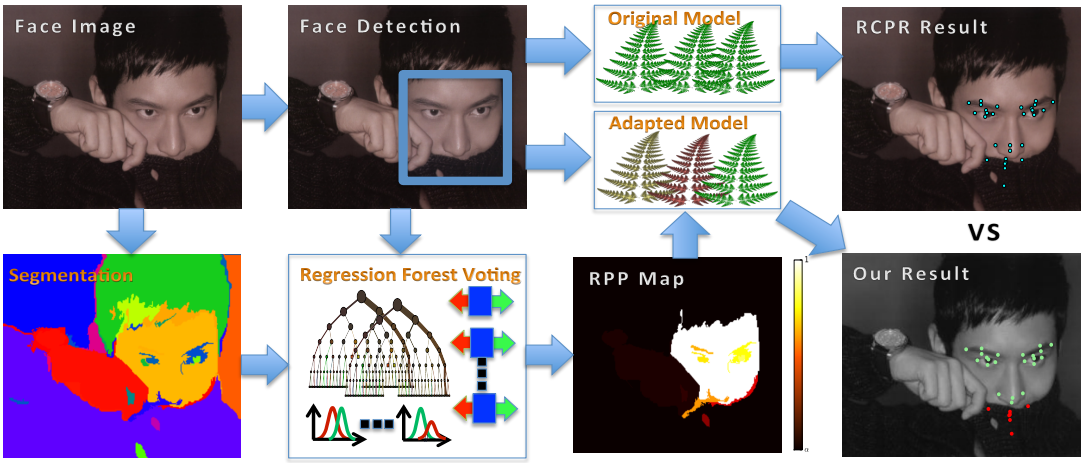


Fig. 4.5 Illustration of the pipeline of the proposed method. Given a test image, we first detect the face and apply segmentation by the graph-based approach in (Felzenszwalb and Huttenlocher, 2004). Based on the face bounding box information and the segmentation result, we employ the local patch based Regression Forest voting method for face alignment and obtain the Regional Predictive Power map with pixel probability from α to 1. We then adapt the state of the art face alignment model, (Robust Cascade Pose Regression (RPP) is used as an example) by putting weights on different weak regressors. The final column shows the results from original RCPR (upper) and the adapted RCPR (lower). Our method is able to localise the landmarks more accurately (especially when occlusion is presented) and reason the occlusion labels of the landmarks (green = unoccluded, red = occluded).

over-segmentation method to partition the image into non-overlapping regions. Then a local regression forests voting based facial feature detection approach is adapted to predict the power of each region connected to the face bounding box. We call this the Regional Predictive Power (RPP) and is essentially a measure of how useful information from a certain region can be for the task of face alignment. The output of this step is a dense RPP map that also indicates the probability of each region belonging to the face. This RPP map is then used along with the original face image for final face alignment using an adapted Cascaded Pose Regression method.

4.2.2 Method

Our method consists of three main parts. In Section 4.2.2 we describe how we use the local Regression Forests voting scheme in order to predict the Regional Predictive Power (RPP) of regions that have resulted from an image (over)segmentation. In Section 4.2.2 we describe how the holistic Cascaded Pose Regression (CPR) face alignment model is adapted

to a more difficult domain, i.e. the domain of occluded images, based on the estimated RPP. Finally, in Section 4.2.2, we present the proposed initialization scheme.

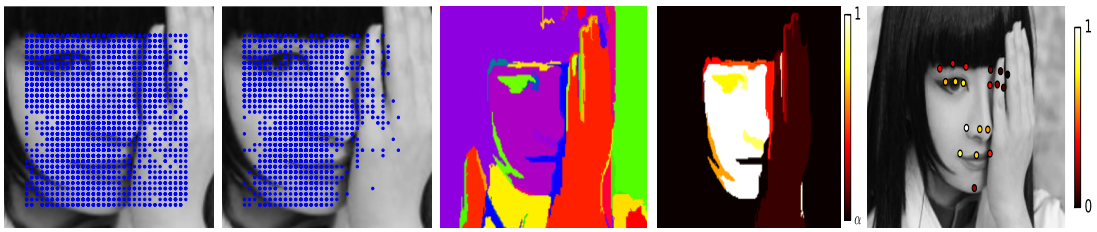
Regional predictive power estimation

It is challenging to directly model the face occlusion due to its unpredictable diversity in realistic conditions. However, the occluders often occupy a continuous region and have different appearance than the face, or are separated from it by intensity edges. We use an over-segmentation and subsequently estimate a score that reflects the power/usefulness of each of the resulting regions in the face alignment task. The score is estimated by analysis of the votes of a local-based Random Forests algorithm, as shown in Fig. 4.6, and is closely related with the probability that the region in question belongs to the face.

We use the efficient graph based segmentation by Felzenszwalb & Huttenlocher (Felzenszwalb and Huttenlocher, 2004), to get a set of regions, which ideally do not span multiple objects (Arbelaez et al., 2011). Let us denote with R the region set and with $r \in R$ a region in that set. The number of regions may vary from image to image. The RPP value of each region is generated in two steps as follows.

Sieving Votes in Regression Forests We build the RPP prediction method based on the Regression Forests (RF) framework for face alignment, proposed in (Dantone et al., 2012b) and in section 2.3. Image patch features that are extracted at several image locations cast votes for the localisation of facial landmarks. As stated in Chapter 2.3, not all the votes from RF are reliable. Therefore, we in Chapter 2.3 propose to use a bank of sieves to remove unreliable votes based on the consistency by which they vote for the location of the face center.

More specifically, a set of patches is extracted from an input image I. Let us denote with V the resulting set of votes and by V_l the subset of the votes that are associated with the landmark l. Clearly, $V = V_1 \cup V_2 \cup \ldots \cup V_L$, where L is the number of landmarks detected by RF. Let us denote by V^r the set of votes that are associated with patches extracted within the region r. Each voting element $v = (\Delta_v, \omega_v, \Delta_v^o, \omega_v^o)$ consists of two types of voting information: one (Δ_v, ω_v) to a facial landmark and the other (Δ_v^o, ω_v^o) to a latent variable, i.e. the face center. Δ_v and ω_v are respectively the offset and the corresponding weight of the vote. (Δ_v^o, ω_v^o) are similarly defined. The face center is localised by using the votes associated with all the landmarks (that is the votes from all image patches); this leads to a robust estimation of its location. Let us denote the estimated face center by $\hat{y}^o$ and assume a voting element v casts a vote at $y_v^o = y_v + \Delta_v^o$ with y_v the image location at which the voting element



(a) (b) (c) (d) (e)

Fig. 4.6 Regression Forests (RF) voting based Region Predictive Power (RPP) estimation. (a) shows the original votes distribution inside the face bounding box, similar dense for both the face region and occlusion region. (b) shows the distribution after the face center sieving as in section 2.3. As can be seen, many invalid votes from the non-face parts are effectively removed, which is a strong cue to predictive the RPP. (c) is the over-segmentation result. (d) shows the RPP map, i.e., the p_r in Eq. 4.8, calculated over each region of the segmentation. (e) is the detection result from the local RF model with the color varies according to the reliability of the landmark estimation, described in Section 4.2.2.

is extracted from, the sieving works as follows:

$$\omega_v := \omega_v \cdot \delta(f|y_v^o - \hat{y}^o| > \lambda^o) \tag{4.7}$$

where $f(\cdot)$ is a transform that converts a distance measure in the range $(\inf, 0]$ to a proximity measure in the range $(0, 1]$ by negative exponential function. λ^o is a threshold. Sieving can be interpreted as a filter that rejects the voting elements whose votes for the face center are far from the estimated center. The set associated with the landmark l and region r after the face center sieving is denoted by $\bar{V}_l$ and $\bar{V}^r$ respectively.

This procedure has been applied to effectively remove the *invalid* votes for facial feature detection. We adopt a similar idea in this work a) for estimating the predictive power of each segmented region as well as b) for estimating the reliability by which each of the facial landmarks is localised by the local-based RF.

RPP Estimation It is difficult to pose the RPP estimation as a supervised classification problem as it is intractable to generate all types of occlusions. Here we take a unsupervised approach that estimate RPP from a set of features based on the region statistics and vote confidence. Specifically, we utilize the votes confidence calculated by the votes sieving procedure. Similarly to section 2.3, we extract features directly from the voting maps as follows:

- $x_r^1 = \frac{\sum_{v \in \bar{V}^r} \omega_v}{\sum_{v \in V^r} \omega_v}$. This is the ratio of the sum of the vote weights in the segmented region r after and before the face center sieve is applied.

- $x_r^2 = U_r$. This is the area size of the region in pixels.

- $x_r^3 = \frac{U_r^{\text{box}}}{U_r}$. This is the fraction of the region that lies inside the face bounding box. U_r^{box} is the area of the region that lies inside the face bounding box. Roughly speaking, the smaller x_r^3 is, the more likely it is that r is an external object, i.e., an occluder of the face. In the example shown in Fig. 4.6, a large proportion of the hand region lies outside the bounding box, and therefore its RPP value is very low.

Give these features, we propose a rule-based method for calculating the RPP as follows. First, we identify the largest most likely face region. We do so, by selecting the M larger regions inside the bounding box and assume that at least one of them belongs to the face. This is a reasonable assumption in real scenarios. From those M regions we select the one with the highest x_r^1 and put it in a set R^0. We then put in R^0 tiny regions, i.e. that satisfy $x_r^2 < \tau$ (where τ is to 50) and set the RPP of all regions in R^0 to 1. The predictive power of all the other regions is estimated based on two strong cues: 1) the more inconsistent votes from one region, the lower RPP; 2) the bigger proportion of one region appears outside the face bounding box, the lower RPP. Formally, the RPP p_r of region r is defined as follows:

$$p_r = \begin{cases} 1 & \text{if } r \in R^0 \\ \alpha + (1-\alpha) x_r^1 x_r^3 & \text{if } r \in R \setminus R^0 \end{cases} . \tag{4.8}$$

The product, $x_r^1 x_r^3$ is normalized to the range of $[0,1]$ in the set of $R \setminus R^0$ and is the main feature used for RPP estimation. The parameter α is the lower bound of the RPP, that is, the range of the RPP is $[\alpha\ 1]$. We empirically set it to 0.2 and will discuss the sensitivity with respect to it in the experimental section.

Face alignment model adaptation with regional predictive power

In this section, we will describe how the above RPP information is used to adapt the Cascaded Pose Regression face alignment model in the presence of un-modeled occlusions. The Cascaded Pose Regression framework has been shown to be effective and accurate in estimating the location of face landmarks (Cao et al., 2012; Dollár et al., 2010) but is sensitive to occlusions (Burgos-Artizzu et al., 2013). Inspired by the weighted mean voting scheme proposed in (Burgos-Artizzu et al., 2013), we leverage the region reliability to augment the CPR model such that the joint model is capable of handling occlusion more effectively.

In the CPR framework, at the t-th iteration, the shape estimated at the previous iteration

S^{t-1} is updated based on shape-indexed features $h^t(S^{t-1}, I)$, where I is the image:

$$S^t = S^{t-1} + \Delta S^t \tag{4.9}$$

where ΔS^t is the shape update. As in (Cao et al., 2012) we use two-level cascaded regression, i.e., at each iteration, there are K primitive fern regressors $R^t = (R^t_1, ..., R^t_k, ..., R^t_K)$ that share the same input, namely features that are indexed relative to S^{t-1}, and whose outputs are combined in order to obtain the shape update ΔS^t as follows:

$$\Delta S^t = \sum_{k=1}^{K} R^t_k(h^t(S^{t-1}, I)) \tag{4.10}$$

Note that despite the fact that the image features used by the K weak regressors are indexed relative to the same pose, the K weak regressors are different random ferns, and therefore the actual image features used by each regressor are at different pixel locations for each one. Assuming F features are used by each fern regressor, we denote the image locations used to calculate the features of the k-th regressor as $\mathrm{x}^k = (\mathrm{x}^k_1, ..., \mathrm{x}^k_f, ..., \mathrm{x}^k_{2F})$. In total, $2F$ pixel locations are used to produce F features. In Section 4.2.2 we have calculated the Regional Predictive Power, thus we can directly get the pixel predictive power according to which region it belongs to. The overall predictive power of the $2F$ locations is calculated as the mean value, that is

$$w_k = \frac{1}{2F} \sum_{f=1}^{2F} \sum_{r \in R} p_r \delta(\mathrm{x}^k_f \in r). \tag{4.11}$$

We adapt the regression model of Eq. 4.10 by reweighing the outputs of the K weak regressors by their respective predictive power. The above weight is normalized to $\bar{w}_k = \frac{K}{\sum_{k=1}^{K} w_k} w_k$, then the shape update at the t-th iteration is:

$$\Delta S^t = \sum_{k=1}^{K} \bar{w}_k R^t_k(h^t(S^{t-1}, I)). \tag{4.12}$$

We note that our method can optionally be integrated with (Burgos-Artizzu et al., 2013), which also introduces area-based local regressors (ferns) and can be viewed as the third level regression. In (Burgos-Artizzu et al., 2013), given the face location in an image, the face is divided into a 3×3 grid. Instead of training a single boosted regressor, N regressors are trained and each regressor is allowed to draw features only from 1 of the 9 pre-defined zones. Finally, each of the regressor's proposed updates $\delta S_1, \cdots, \delta S_N$ are combined through

a weighted mean voting, where weight is inversely proportional to the occlusion estimation in the zones from which the regressor drew features. We can combine our RPP estimation with RCPR as follows. For the k-th update at the t-th iteration,

$$\Delta S_k^t = \sum_{n=1}^{N} \bar{w}_k^n \delta S_n^k. \tag{4.13}$$

This is the same form as that of the RCPR but the weight $\bar{w}_k^n$ is directly deduced from the RPP map as in Eq. 4.12 rather than estimated from the previous iteration .

Initialization from local-based model

Existing iterative methods, e.g., the SDM (Xiong and De la Torre, 2013) and CPR (Dollár et al., 2010), depend on initialization and only those initializations that lie within a certain range can converge to the correct solution. However, there is no guarantee that the same face detector is used during the testing and training time. For instance, the SDM is trained based on mean pose deduced from Viola-Jones detector, however, Viola-Jones face detector misses many faces in the COFW dataset due to its heavy occlusion. Here we propose an initialization scheme that uses the estimated landmark locations and their estimated reliability, as those are provided by the local based Regression Forests method. Since the RF-based method is based on local patch features it does not require initialization, thus it is inherently more robust to face bounding box shifts.

Specifically, let us denote that the estimate from the RF method in Section 4.2.2 by $y = (y_1, ..., , y_l, ..., y_L)$. Here, we also estimate the reliability of each landmark, that is, the confidence that the localisation is correct. This differs from most of the face alignment methods. The reliability of a landmark is derived from the votes that are used to localise it and is calculated as follows:

$$s_l = \sum_{v \in \bar{V}_l} \omega_v \Big/ \sum_{v \in V_l} \omega_v \tag{4.14}$$

We then find the L_{com} common landmarks shared by the RF-based model and the RCPR model. Then instead of randomly selecting m shapes from the training set, we search the m nearest neighbors to the shape estimated by the RF. The distance between shapes is calculated as the sum of weighted Euclidian distances of all the common landmarks, where the weights are the given by Eq. 4.14. This weighted distance measure suppresses the impact of the landmarks with large localisation errors. Formally, the distance from the estimated

shape vector y,to another shape y' is given by,

$$d(y,y') = \sum_{l=1}^{L_{com}} s_l ||y_l - y'_l||_2. \tag{4.15}$$

Note that, when calculating the distance, all the shapes are first normalized by procrustes analysis. This distance is used to calculate the m nearest neighbors in the training set - those are used to initialize the cascaded method.

4.2.3 Implementation details

We report the performance of our method on the most challenging datasets, namely, the Caltech Occluded Faces in the Wild (**COFW**) (Burgos-Artizzu et al., 2013) dataset and the 300 Faces in-the-Wild (**300W**) (Sagonas et al., 2013c).

Since 300W only provides the training images for the challenge, thus we follow the experiment setting of (Ren et al., 2014) in order to compare with the recent methods. The training set is split into two parts. More specifically, the training part consists of AFW, the training images of LFPW and the training images of HELEN, with 3148 samples in total. The testing set consists of the test images of LFPW, the test images of HELEN and the images in the iBug set, with 689 samples in total. The test set is further partitioned into Easy-set (LFPW and HELEN test images) and Challenging-set (iBug images).

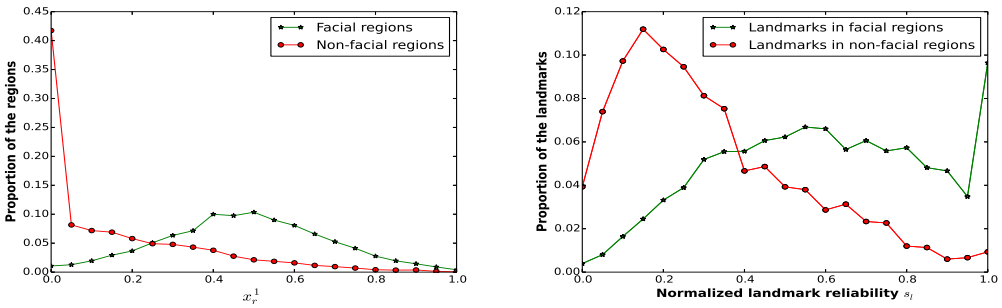
For the local Regression Forests, we use the trained model provided in section 2.3, which is trained on a subset of AFLW (Kostinger et al., 2011) that contains mostly near frontal face images to ensure that the 19 facial landmarks are visible. We use all their default model parameters setting. Given that our adaptation methodology works on those models, it is clear that it does not exploit any training instances or annotations such as the occlusion labels. In our adaptation model, the number of the largest regions, that is the variable M in section 4.2.2, is set to 3. The number of nearest neighbors that are used for initialization, that is, the variable m in section 4.2.2 is set to 5 - this is the default setting for RCPR. The error is measured as a fraction of the interocular distance. We note that in the evaluation process except when explicitly testing the face bounding box shift caused by changing the face detectors in Section 4.2.4, the same face detector is used for both training and testing for fair comparison.

4.2.4 Results

RPP estimation evaluation

We empirically evaluate the performance of the RPP estimation based on the facial area annotation on COFW test images. Note that we do not use the annotation to tune our system during training. We set a threshold, equal to $\tau_{RPP} = \frac{1+\alpha}{2}$, on the RPP map. Regions with RPP value larger than the threshold are considered to be facial regions, and regions with smaller values are considered to be occlusions. Since we have annotated the face region masks for the testing images, we calculate the overlap area ratio inside the face bounding box to measure the performance, $\rho = \frac{A_{PPR} \cap A_{GT}}{A_{PPR} \cup A_{GT}}$. The average ratio is 72.4%, which is surprisingly high, given that the average percentage of area occlusion is 46.2%. We further infer the landmark occlusion state. If the RPP value of the region that one landmark is located in is larger than a threshold τ_{RPP}, the landmark is regarded as visible, and vice versa. For landmark occlusion detection we get a 78/40% precision/recall, which is close to the 80/40% precision/recall reported in (Burgos-Artizzu et al., 2013). We note that in contrast to (Burgos-Artizzu et al., 2013) we do not use occlusion information during training.

Feature analysis



(a) (b)

Fig. 4.7 The distribution of x_r^1 feature (a) and landmark reliability s_l (b) for facial regions and non-facial regions. In (b) the value s_l of one face is normalized in the range between 0 and 1.

In Section 4.2.2 we developed features for RPP computation and reliability metric for landmark localisation. We mainly rely on two features for RPP estimation, i.e. x_r^1 and x_r^3. In order to show the relevance of x_r^1, based on the face mask annotation, we plot the histogram of feature values for the face-regions and non-face regions, respectively, in Fig. 4.7(a). The p.d.f of x_r^1 in non-facial regions decreases gradually. On the contrary, the p.d.f of x_r^1 in

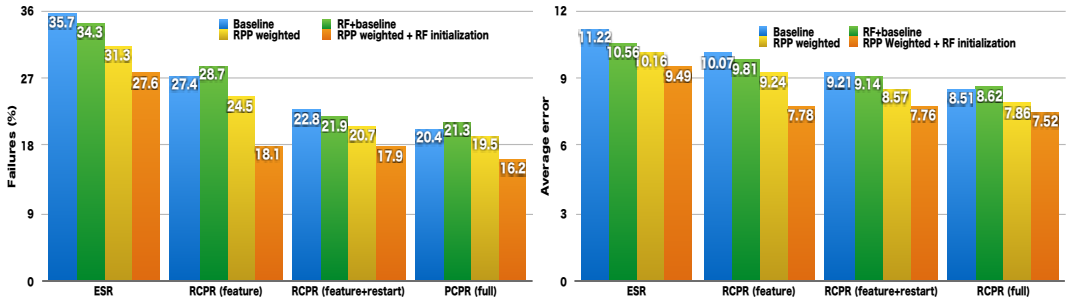


Fig. 4.8 Results on COFW, compared to CPR-family approaches (Burgos-Artizzu et al., 2013; Cao et al., 2012).

facial regions peaks at around 0.5. In Fig. 4.7(b) we plot the histogram of the landmarks reliability s_l, defined in Eq. 4.14, from non-occluded and occluded face regions. We see that the reliability of most landmarks under occlusion tend to be lower than the reliability of the visible landmarks.

Face alignment evaluation on COFW

Here we evaluate the contribution of each component of the proposed method. We take four models from the CPR family as baseline methods: 1) the Explicit Shape Regression(ESR) (Cao et al., 2012); 2) the feature only version of RCPR (Burgos-Artizzu et al., 2013); 3) the RCPR with feature and smart restart (Burgos-Artizzu et al., 2013); 4) the full version of the RCPR. All of them are trained on the COFW training images with the same settings except the RCPR (full) which has used the landmark visibility labels during training. In the experimental comparison, **RF+baseline** is the direct combination of the RF sieving (Chapter 2.3) and the baseline method, i.e. the output of RF sieving is used to find non-weighted nearest neighbouring shapes (all s_l in Eq. 4.14 are set to 1) to initialize the baseline methods. **RPP weighted** applies only the RPP weighting as described in Eq. 4.12 and **RPP weighted+RF initialization** is our full method. Note that RCPR (full) uses N visually different regressors at each iteration. For a fair comparison, we *replace* their predicted weight with our RPP based weight for each visually different regressor. For methods not based on RF initialization we use 5 random initializations, that are the same for all methods. For the RF-based initialization methods, we replace the 5 initializations with the searched results. For the face images that need *smart restart*, the initializations in restart are all randomly generated. We repeat this process 4 times and report the average performance in terms of the proportion of failures and average errors, similar to (Burgos-Artizzu et al., 2013). The number of restarts in the second round is also recorded as it is an important indicator of the efficiency. The results are shown in Fig. 4.8. We also compare to the recent methods which provide code

on the common landmarks of the COFW test images, as shown in Fig. 4.9.

We can draw the following conclusions from the results: 1) the direct combination (**RF+baseline**) does not perform better than the baseline method; 2) the weighted models improve all the baseline methods in the CPR family, at an average mean error reduction of 0.8 and a decrease of failure rate of 2.6%; 3) it is worthy to note that the RPP based weights are even more effective than the original *learned* weights used in the RCPR (full) model, with a failure case decreased 1% and a mean error decrease of 0.65; 4) the proposed initialization scheme is very effective and further decreases the mean error by 0.8 and the failure cases by 4%. 5) The smart restart has less impact when our proposed initialization scheme is applied. The number of restarts decreases from 200 to 30 among the 507 images, which means much fewer instances (85% less) are required to restart the initializations (Burgos-Artizzu et al., 2013). The comparison to other state of the art methods on COFW is shown in Fig. 4.9, where the proposed method, that is built on top of RCPR (feature only), shows superior performance. Some examples are shown in Fig. 4.11.

We compare to the recent Hierarchical Deformable Part Model (HDPM) (Ghiasi and Fowlkes, 2014) using their best performing setting, i.e. the model is trained on HELEN images and tested on the COFW test set. The average error and failure rate of our method is 0.0713 and 12.51% while that of HDPM is 0.0746 and 13.24%, respectively.

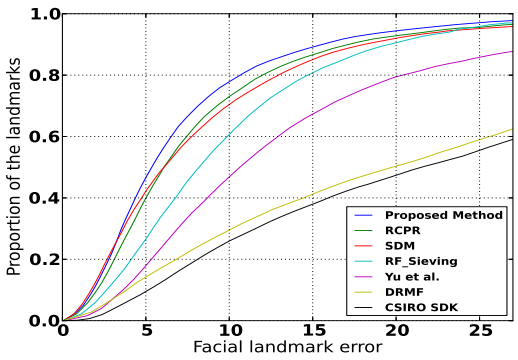
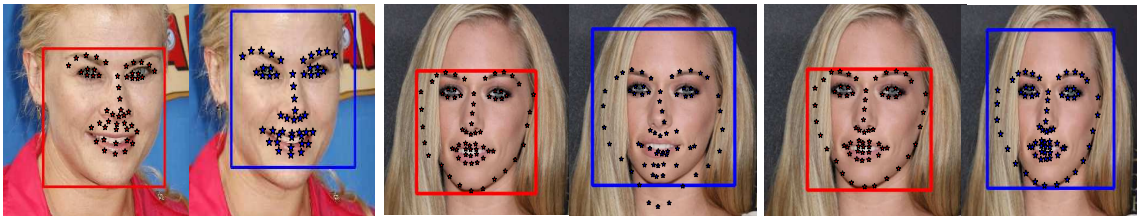


Fig. 4.9 Comparison to the recent methods, SDM (Xiong and De la Torre, 2013), RCPR (Burgos-Artizzu et al., 2013), RF_Sieving in section 2.3, method of Yu et al. (Yu et al., 2013a), DRMF (Asthana et al., 2013), and CSRIO SDK (Cox et al., 2013) on COFW test images for their common 16 facial landmarks. For the DRMF, the pre-computed face bounding box model is used since the tree-based method does not work on such images.

In the proposed RPP model, there is one parameter α that influences the facial landmark localisation. We increase its value from 0 to 1 with a step of 0.1 for the PCPR (feature+restart) model. The corresponding failure cases are $[23.4, 21.3, 20.7, 20.5, 20.7, 21.1, 21.4, 22.7, 22.6, 22.7, 22.8]$. When α is set to 0, the result



(a) SDM(Xiong and De la Torre, 2013) (↑ 20%) (b) RCPR(Burgos-Artizzu et al., 2013) (↑ 7%) (c) Our method (↑ 4%)

Fig. 4.10 Example results based on Viola-Jones face detector (blue) and 300-W face detector (red). SDM is trained based on Viola-Jones face detection and the other two are trained on 300-W face detection. The number under each pair shows increase of failure cases when face detection changes from one to the other.

is the worst, when α lies between 0.1 and 0.5, the performance is stable and when α becomes larger than 0.5, the performance approaches gradually to the baseline method, i.e., the model with equal weights. We set the value to 0.2 in all our experiment. Its value can be set by cross validation in practice.

Face alignment evaluation on 300W dataset

Table 4.2 300-W dataset (68 landmarks).

Method	Full-set	Easy-set	Challenging-set
ESR(Cao et al., 2012)	7.58	5.28	17.00
SDM(Xiong and De la Torre, 2013)	7.52	5.60	15.40
LBF fast(Ren et al., 2014)	7.37	5.38	15.50
LBF(Ren et al., 2014)	6.32	4.95	11.98
RCPR(Burgos-Artizzu et al., 2013) (baseline)	7.54	5.67	15.50
Our method	6.69	5.50	11.57

Table 4.3 300-W dataset (49 landmarks).

Method	Full-set	Easy-set	Challenging-set
IFA(Asthana et al., 2014)	7.48	5.58	15.30
SDM(Xiong and De la Torre, 2013)	7.06	5.56	13.22
RCPR(Burgos-Artizzu et al., 2013) (baseline)	7.20	5.47	14.28
Our method	6.57	5.40	11.40

On the 300-W dataset we compare our proposed method with the most competitive methods including the Supervised Descent Method (SDM) (Xiong and De la Torre, 2013),

the ESR[2] (Cao et al., 2012), the Incremental Face Alignment (IFA) (Asthana et al., 2014) and the RCPR(Burgos-Artizzu et al., 2013). We use the full-RCPR version but we do not use any occlusion labels. For each of the regressor in Eq. 4.13, we treat them equally during the training stage and set the weight to $\frac{1}{N}$. This is equivalent to treat all landmarks are visible. We take this as the baseline for adaptation as this gives us the best results compared to other RCPR variants. We first make the comparison as shown in Table 4.2 where the results of SDM, ESR, LBF and LBF-fast are quoted from (Ren et al., 2014). We train the baseline RCPR model on the same training set for a fair comparison. As can be seen, although we only have comparable results to LBF, our results are better than the rest of the models. We note that LBF needs to train hundreds of thousands of trees while in our method, both the local Regression Forest and the RCPR model is quite easy to train and the model size is much smaller. The improvement over the baseline RCPR model validates the effectiveness of our proposed method. We then compare to IFA and SDM in Table 4.3, as they show the state of the art results and have available test code. We train the baseline RCRP model on the Multi-PIE+LFPW (similar to the SDM model according the description of the paper) for localising the 49 inner facial landmarks. Then we apply our adaptation method on it. As can be seen, though the baseline RCPR method fails to compete the IFA and SDM, our method improves it clearly and shows better performance. From the two comparison we also note that the superior performance of our method on the Challenging subset is more significant, which is as expected since those images contain much more occlusions.

Face bounding box shifts

We evaluate the effect of face bounding box changes that is caused by different face detectors on the esay set of 300-W (LFPW and HELEN test images). As shown in Fig. 4.10, when the face bounding box changes, the performance of the cascaded methods changes significantly. The failure cases of the SDM method increases by 20% on average when the face bounding box of the test images changes from Viola-Jones face detector to 300-W face detector while that of the RCPR increases by 7% when the face bounding box changes the other way. The fact that the increase in failure of the SDM method is higher than that of the PCPR is probably due to their difference in initialization methodology, since the SDM only calculates one pose from the bounding box for initialization while the RCPR randomly selects 5 from the training instances. By using our proposed initialization scheme, the increase is minor (4%), around a half of the baseline RCPR method.

[2]The result might be different from that in Section 4.2.4, where the re-implementation source code is used.

Fig. 4.11 Example results from COFW (first two rows) and LFPW and HELEN (last two rows), including landmarks detection results (upper) and the corresponding RPP map (lower). See Fig. 4.5 for color map definition.

Run-time

We record the run-time performance on a standard 3.30GHz CPU machine. For the COFW test images, the fps of the three components (segmentation (c++), Regression Forest (c++) and CPR (Matlab)) of our proposed method is 12, 17 and 11, respectively, and the overall speed is 4 FPS, that is a bit faster than the RCPR (full) method, and much faster than the HDPM (Ghiasi and Fowlkes, 2014) (0.03FPS). On the LFPW and HELEN, the speed is 3.3 fps and 1 fps respectively while the segmentation takes longer time when the image becomes larger. Applying the segmentation only at a region of interest surrounding the face bounding box instead of the whole image can make our method more efficient.

4.3 Summary

In this chapter, we present two methods specifically for face alignment under occlusion.

First, we present a structured semi-supervised forest model for joint face mask reasoning and facial landmark localisation. We augmented a portion of training images with densely manually-labelled face masks that are used for structured output learning, based on our proposed structural information gain criterion. Experiments show that the proposed framework achieves accurate and spatial consistent face mask prediction, which further assists the landmark localisation. We have focused on comparing to the Regression Forests based method and show competitive performance in both tasks. As our method is still a local facial feature detection, we believe that it could be incorporated into a range of model matching frameworks for facial landmarks localisation performance boost.

Second, we present a method for face alignment model adaptation, relying on Regional Predictive Power (RPP), based on image segmentation. We achieve the state of the art results for face alignment in challenging databases, despite the fact that we do not need additional annotation at the training stage to tune the model.

These two methods also show the efficacy on facial region prediction in pixel level or region level, something that can have applications in face analysis in real world applications such as face verification and facial expression recognition. To the best of our knowledge, this is the first work that tries to localise facial landmarks and predict dense facial mask in a joint framework.

This work also raises a few interesting problems. First, with the rapid progress of face alignment, there is a demand of more advanced face detectors that can work better in an unconstrained environment, since even the state of the art face detectors fail under heavy occlusion. Second, while most of the current methods work quite well on images with minor partial occlusion in a very fast speed but struggle under occlusion, developing a method based on the difficulty level of the test image to select a proper model is useful for practical applications.

Chapter 5

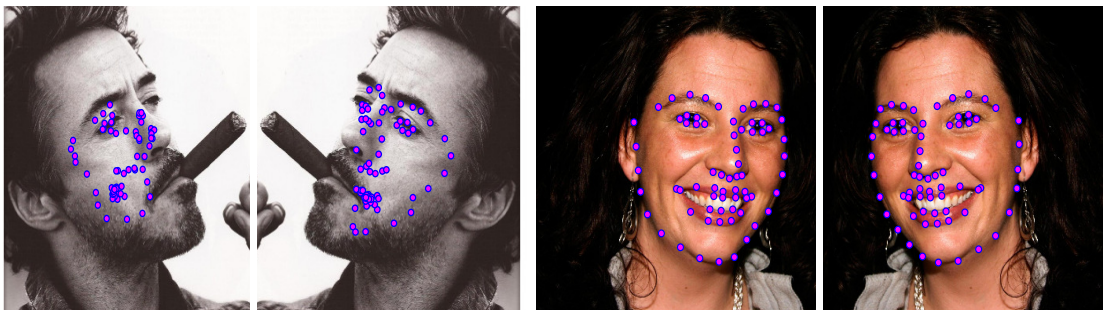
Face Alignment Mirrorability

In previous chapters, we have presented a set of face alignment methods, from local based ones to holistic based ones, and addressed a series of challenges such as head poses variation, large number of facial landmarks, unreliable initializations, heavy occlusions etc. In this chapter, we evaluate the approaches for face alignment from a different perspective. We introduce the concept mirrorability, as the ability of a model/algorithm to preserve the mirror symmetry when applied on an image and its mirror image. We focus on an image mirror transform in this thesis because 1) most of the evaluated models are trained on both the original and the mirror images, 2) the image features like HOG are symmetric under the mirror transform and 3) other transforms are in continuous space i.e. it is difficult to define how much to rotate the image.

We evaluate the mirrorability of several state of the art algorithms in face alignment and present several interesting findings. We show two interesting applications - in the first it is used to guide the selection of difficult samples and in the second to give feedback to a Cascaded Pose Regression method for improving its performance on face alignment.

5.1 Introduction

The evolution of mirror (bilateral) symmetry has profoundly impacted animal evolution (Finnerty, 2005). As a consequence, the overwhelming majority of modern animals (>99%), including humans, exhibit mirror symmetry. As shown in Fig. 5.1, the mirror of an image depicting such objects shows a meaningful version of the same objects. Taking face images as a concrete example, a mirrored version of a face image is perceived as the same face. In recent year, face alignment has made significant progress and several methods have reported close-to-human performance. Most of these methods augment the training set by mirroring the positive training samples. However, are these models able to give symmetric results on



(a) Mirror error 0.6. (b) Mirror error 0.02.

Fig. 5.1 Example pairs of localisation results on original (left) and mirror (right) images. The first column (a) shows large mirror error and the second (b) small mirror error. Can we evaluate the performance without knowing the ground truth?

a mirror images during testing?

In order to quantitatively measure the mirrorability we introduce a measure called mirror error, which is defined as the difference between the detection result on an image and the mirror of detection result on its mirror image. We evaluate the mirrorability of several state of the art algorithms in face alignment on several datasets. One would expect that a model that has been trained on a dataset augmented with mirror images to give similar results on an image and its mirrored version. However, as can be seen in Fig. 5.1 first column, several state of the art methods sometimes struggle to give symmetric results. For some samples, the mirror error is quite large. By looking at the mirrorability of different approaches in face alignment, we arrive at three interesting findings. First, most of the models struggle to preserve the mirrorability - the mirror error is present and sometimes significant; Second, the low mirrorability is not likely to be caused by training or testing sample bias - the training sets are augmented with mirrored images; Third, the mirror error of the samples is highly correlated with the corresponding ground truth error.

This last finding is significant since one of the *nice* properties of the proposed mirror error is that it is calculated 'blindly', i.e., without using the ground truth. We rely on this property in order to show two examples of how it could be used in practice. In the first one, the mirror error is used as a guide for difficult sample selection in unlabelled data and in the second one it is used to provide feedback on a cascaded pose regression method for face alignment. In the former application, the samples selected based on the mirror error have shown high consistency across different methods and high consistency with the difficult samples selected based on the ground truth alignment error. In the latter application, the feedback mechanism is used in a multiple initializations scheme in order to detect failures - this leads to large improvements and state of the art results in face alignment.

To summarize, in this work we make the following contributions:

- To the best of our knowledge, we are the first to look into the mirror symmetric performance of face alignment models.
- We introduce the concept of mirrorability and show how the corresponding measure, called mirror error, that we propose can be used in evaluating face alignment methods.
- We evaluate the mirrorability of several algorithms in face alignment and report several interesting findings on the mirrorability.
- We show two applications of the mirrorability in the domain of face alignment.

5.2 Related Work

As a method that estimates the quality of the output of a vision system, our method is related to works like the meta-recognition (Scheirer et al., 2011), face recognition score analysis (Wang et al., 2007) and the recent failure alert (Zhang et al., 2014c) for failure prediction. Our method differs from those works in two prominent aspects (1) we focus on a fine-grained object part localisation problem while they focus on instance level recognition or detection. (2) we do not train any additional models for evaluation while all those methods rely on meta-systems. In the specific application of evaluating the performance of Human Pose Estimation, (Jammalamadaka et al., 2012) proposed an evaluation algorithm, however, again such an evaluation requires a meta model and it only works for that specific application.

Our method is also very different from object/feature detection methods that exploit mirror symmetry as a constraint in model building (Loy and Eklundh, 2006; Tsogkas and Kokkinos, 2012). We note that our model does not assume that the detected object or shape appears symmetrically in an image - such an assumption clearly does not hold true for the deformable face objects that we are dealing with. None of the methods that we have exploited in this work explicitly used the appearance symmetry in model learning. Our method only utilizes the mirror symmetry property to map the object parts between the original and mirror images.

Developing a transformation invariant vision system has drawn much attention in the last decades. Examples are the rotation invariant face detection method (Rowley et al., 1998) and the scale invariant feature transform (SIFT) (Lowe, 1999), which handle efficiently several transformations including the mirror transformation. Recently, Gens and Domingos proposed the Deep Symmetry Networks (Gens and Domingos, 2014) that use symmetry

groups to represent variations - it is unclear though how the proposed method can be applied for object part localisation. Szegedy *et al.* (Szegedy et al., 2013) has studied some intriguing properties of neural networks when dealing with certain artificial perturbations. Our method focuses on examining the performance of object part localisation methods on one of the simplest transforms, i.e. mirror transformation, and drawing useful conclusions.

5.3 Mirrorability in Face Alignment

5.3.1 Mirrorability concepts and definitions

We define mirrorability as the ability of a model/algorithm to preserve the mirror symmetry when applied on an image and its mirror image. In order to quantify it we introduce a measure called mirror error that is defined as the difference between a detection result on an image and the mirror of the result on its mirror image. Specifically, let us denote the shape of an object, for example a human or a face, by a set of K points, $S = \{\mathrm{y}_k\}_{k=1}^{K}$, where y_k are the coordinates of the k-th point/part. The detection result on the original image is denoted by ${}^{q}S = \{{}^{\mathrm{q}}\mathrm{y}_k\}_{k=1}^{K}$ and the detection result on the mirror image is denoted by ${}^{p}S = \{{}^{\mathrm{p}}\mathrm{y}_k\}_{k=1}^{K}$. The mirror transformation of ${}^{p}S$ to the original image is denoted by ${}^{\mathrm{p}\to\mathrm{q}}\mathrm{S} = \{{}^{\mathrm{p}\to\mathrm{q}}\mathrm{x}_k\}_{k=1}^{K}$, where ${}^{\mathrm{p}\to\mathrm{q}}\mathrm{y}_k$ denotes the mirror result of the k-th part on the original image. Generally, a different index k' is used on the mirror image (e.g. a left eye in an image becomes a right eye in the mirror image). Therefore, the transformation consists of image coordinates transform and the part index mirror transform ($k' \to k$). The image coordinate transform is applied on the horizontal coordinate, that is ${}^{\mathrm{p}}y_k = \mathrm{w}_I - {}^{\mathrm{q}}y_k$, where w_I is the width of the image I and ${}^{\mathrm{p}}y_k$ is the x coordinate of the k point in the mirror image. The index re-assignment is based on the the mirror symmetric structure of a specific object, with an one-to-one mapping list where, for example, the left eye index is mapped to the right eye index. Formally, the mirror error of the k landmark (body joint or facial point) is defined as $||{}^{\mathrm{q}}\mathrm{y}_k - {}^{\mathrm{p}\to\mathrm{q}}\mathrm{y}_k||$, and the sample-wise mirror error as:

$$e_{\mathrm{m}} = \frac{1}{K}\sum_{k=1}^{K} ||{}^{\mathrm{q}}\mathrm{y}_k - {}^{\mathrm{p}\to\mathrm{q}}\mathrm{y}_k|| \tag{5.1}$$

The mirror error that is defined in the above equation has the following properties: First, a high mirror error reflects low mirrorability and vice visa; Second, it is symmetric, i.e., given a pair of mirror images it makes no difference which is considered to be the original; Third, and importantly, calculating the mirror error does not require ground truth information.

In a similar way we calculate the ground truth localisation error ${}^{\mathrm{q}}e_{\mathrm{a}}$ as the difference

between the detected locations and the ground truth locations of the facial landmarks. In order to be consistent and distinguish it from the mirror error we call it the alignment error. Formally,

$$ {}^{\mathrm{q}}e_{\mathrm{a}} = \frac{1}{K}\sum_{k=1}^{K} ||{}^{\mathrm{q}}\mathrm{y}_k - {}^{\mathrm{gt}}\mathrm{y}_k|| \tag{5.2} $$

where ${}^{\mathrm{gt}}\mathrm{y}_k$ is the ground truth location of the k-th point. In a similar way, we define the alignment error ${}^{\mathrm{p}}e_{\mathrm{a}}$ on the mirror image of the test sample. For simplicity in what follows when we use the term of alignment error e_a, we mean the alignment error in the original image.

Both Eq. 5.1 and Eq. 5.2 are absolute errors. In order to keep our analysis invariant to the size of the object in each image, we normalize them by the face size, i.e. s, the size of the face.

5.3.2 Experiments

In this section we look into the mirrorability of face alignment methods and how their error is correlated to the mirror error.

Experiment setting For our analysis we focus on the most challenging datasets, namely the 300W. We perform our analysis on a test set that comprises of the test images from HELEN (330 images), LFPW (224 images) and the images in the iBug subset (135 images), that is 689 images in total. The images in the iBug subset are extremely challenging due to the large head pose variations, faces that are partially outside the image and heavy occlusions. The test images are flipped horizontally to get the mirror images. We evaluate the performance of several recent state of the art methods, namely the Supervised Descent Method (**SDM**) (Xiong and De la Torre, 2013), the Robust Cascaded Pose Regression (**RCPR**) (Burgos-Artizzu et al., 2013), the Incremental Face Alignment (**IFA**) (Asthana et al., 2014) and the Gaussian-Newton Deformable Part Model (**GN-DPM**) (Tzimiropoulos and Pantic, 2014). For SDM, IFA and GN-DPM, only the trained models and the code for testing is available - we use those to directly apply them on the test images. As stated in the corresponding papers, the IFA and GN-DPM were trained on the 300W dataset and the SDM model was trained using a much larger dataset. SDM, IFA and GN-DPM only detect the 49 inner facial points - our analysis on those methods is therefore based on those points only. For RCPR, for which the code for training is available, we retrain the model on the training images of 300W for the full 68 facial points mark-up. All those methods build on the result of a face detector - since most of them are sensitive to initialization, we carefully choose the

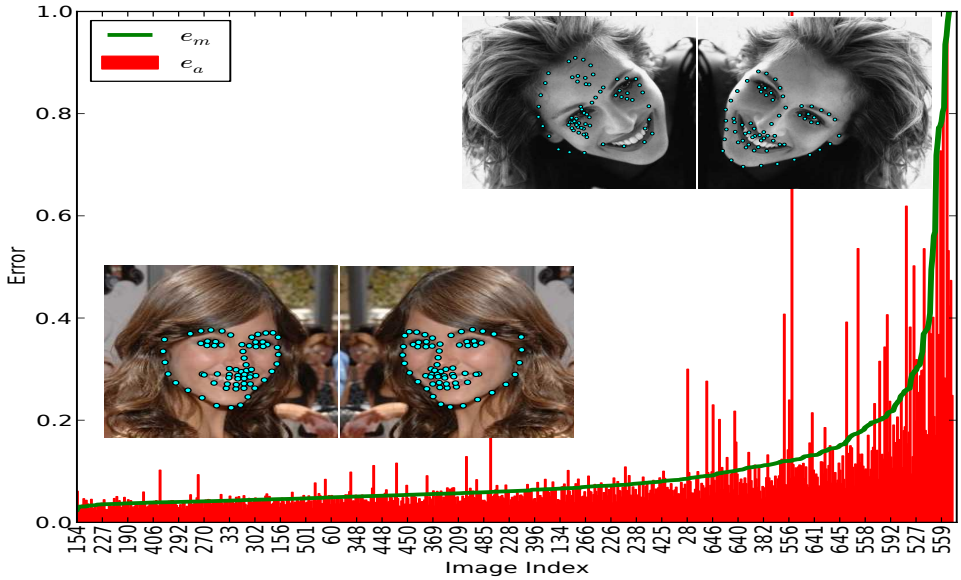


Fig. 5.2 Mirror error and alignment error of RCPR (Burgos-Artizzu et al., 2013) on 300W test images. Results are calculated over 68 facial points.

right face detector for each one to get the best performance. More specifically, for the IFA and GN-DPM we use the 300W face bounding boxes and for SDM and RCPR we use the Viola-Jones bounding boxes, that is for each method we used the detector that it used during training. For the methods that use the Viola-Jones bounding boxes, we checked manually to verify that the detection is correct - for those face images on which the Viola-Jones face detector fails, we adjust the 300W bounding box to roughly approximate the Viola-Jones bounding box.

Mirrorability We calculated the mirror error and the alignment error for each of the 689 test samples in 300W for SDM, IFA, GN-DPM and RCPR. In Fig. 5.3 and Fig. 5.2 we show the errors for two of the algorithms, i.e., the GN-DPM and the RCPR. The former is a representative local-based method and the latter a representative holistic-based method. Similar results were obtained for SDM and IFA. In each figure, two pairs of example images are shown - one with low mirror error (lower left corner) and one with large mirror error (upper right corner). We sort the sample-wise alignment error in ascending order and plot it together with the corresponding sample mirror error. It is clear that although GN-DPM and the RCPR work in a very different way, for both the mirror error tends to increase as the alignment error increases. There are a few impulses in the lower range of the red curve, i.e., low ${}^{\mathrm{q}}e_a$ and high e_m. This means that although the algorithm has small alignment error on the original samples it has large error on the mirror images, i.e., ${}^{\mathrm{q}}e_a$ is high. There are three cases that result in high mirror error: 1) low ${}^{\mathrm{q}}e_a$ and high ${}^{\mathrm{p}}e_a$; 2) high ${}^{\mathrm{q}}e_a$ and low ${}^{\mathrm{p}}e_a$

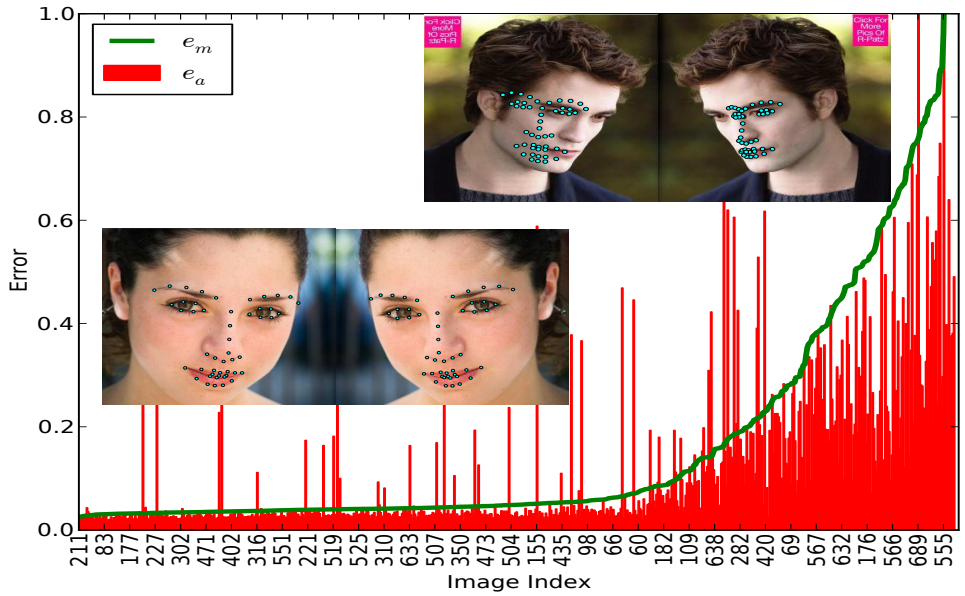


Fig. 5.3 Mirror error and alignment error of GN-DPM (Tzimiropoulos and Pantic, 2014) on 300W test images. Results are calculated over 49 inner facial points.

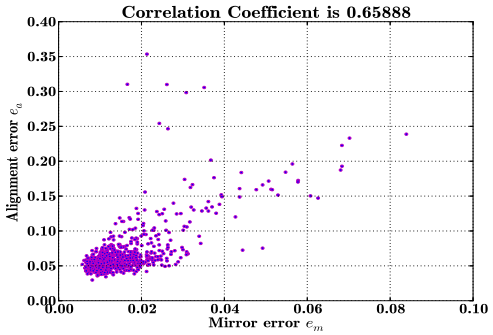
(shown in Fig. 5.2 upper right corner); 3) high ${}^{q}e_a$ and high ${}^{p}e_a$ (shown in Fig. 5.3 upper right corner). Finally, in order to quantify this insight, we present the correlation between the mirror error and the alignment error in Fig. 5.4. In all of the four methods there is a strong correlation between the mirror error and the alignment error with correlation coefficients ranging from 0.64 to 0.74 - these are very high.

5.4 Mirrorability Applications

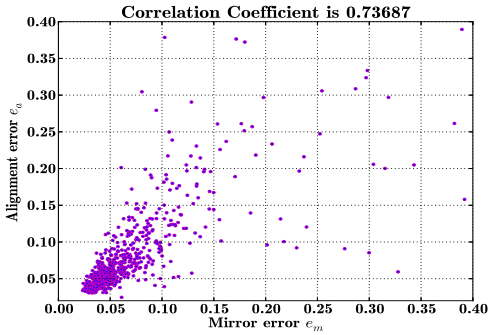
In the previous sections we have shown that one of the nice properties of the mirror error is that it is strongly correlated with the face alignment error, that is with the ground truth error. In this section we show how it can be used in two practical applications, namely for selecting difficult samples and for providing feedback in a cascaded face alignment method.

5.4.1 Difficult samples selection

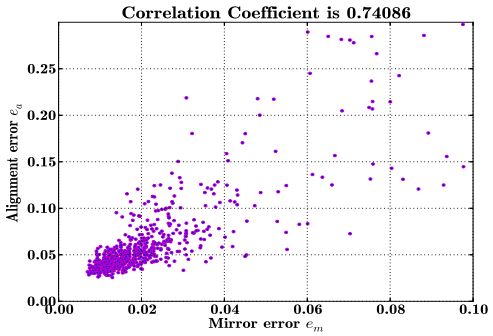
For any computer vision task, including face alignment, it is generally accepted that some samples are relatively more difficult than others, that is the error of the algorithm on them is higher. However, it is very difficult to estimate a measure of how well the algorithm has performed on a given sample without knowledge of the ground truth. Such a measure would be very useful, for example in order to select a proper alignment model for a given dataset or to select which samples to annotate in an Active Learning scheme. Here, we show how the



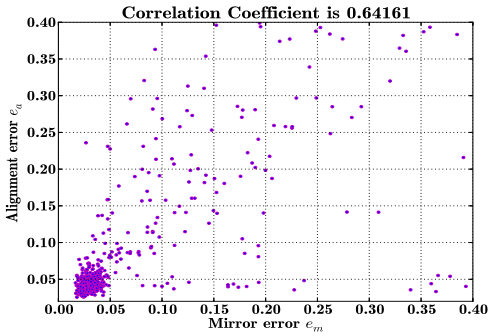
(a) SDM (Xiong and De la Torre, 2013), 49P



(b) RCPR (Burgos-Artizzu et al., 2013), 68P

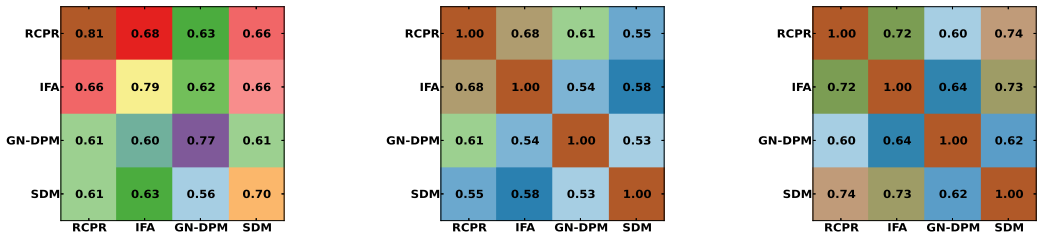


(c) IFA (Asthana et al., 2014), 49P



(d) GN-DPM (Tzimiropoulos and Pantic, 2014), 49P

Fig. 5.4 Correlation between the alignment error and the mirror error of various state of the art face alignment methods. The correlation coefficients are shown above the figures.



(a) ρ of $\mathcal{S}_{e_a} \Leftrightarrow \mathcal{S}_{e_m}$. (b) ρ of $\mathcal{S}_{e_m} \Leftrightarrow \mathcal{S}_{e_m}$. (c) ρ of $\mathcal{S}_{e_a} \Leftrightarrow \mathcal{S}_{e_a}$.

Fig. 5.5 Consistency measure of 'difficult' samples detection, with $M = 150$.

mirror error can be used for selecting difficult samples in the problem of face alignment. In order to do so we apply several methods (IFA, SDM, GN-DPM, RCPR) on the test images of the 300W and get the detection results. Then we sort the normalized mirror error e_m in descending order and select the first M samples as being the most difficult ones. We denote this set as $\mathcal{S}_{e_m}$.

In order to evaluate whether the samples that we have selected in this way are truly 'difficult' we measure the similarity between the set containing those M selected samples and the set $\mathcal{S}_{e_a}$ that contains the M samples that have the largest alignment error e_a for each method. We use a measure that we call consistency which we define as the fraction of the common samples between the two sets, that is

$$\rho = \frac{|\mathcal{S}_1 \cap \mathcal{S}_2|}{M} \tag{5.3}$$

where $|\mathcal{S}_1 \cap \mathcal{S}_2|$ is the size of the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. For each method i, we calculate two sets each containing M samples, i.e., $\mathcal{S}^i_{e_m}$ and $\mathcal{S}^i_{e_a}$. We set the value of M to 150. The chance rate is $\frac{M}{N}$, where M is the number of selected and N is the size of the dataset - in our case is $\frac{150}{689} \approx 0.22$.

The pairwise consistency rate matrix of $\mathcal{S}^i_{e_m}$ and $\mathcal{S}^i_{e_a}$ is shown in Fig. 5.5a, where in a certain row we show the consistency between the $\mathcal{S}^i_{e_m}$ of a certain method with the $\mathcal{S}^i_{e_a}$ of all methods, including the method itself. Note that the diagonal does not contain ones, since $\mathcal{S}^i_{e_m}$ are the M samples with the highest mirror error and $\mathcal{S}^i_{e_a}$ the M samples with the highest alignment error. As it can be seen, the consistency between the two sets of samples for a specific method (i.e., the diagonal values) are all above 0.7 - the highest is 0.81 for RCPR. More interestingly, the consistency across different methods, i.e., the M samples selected according to e_a for a method in a certain row and the M samples selected according to e_m in a certain column is high, with values ranging from 0.56 to 0.68. This shows that the samples that we have selected are truly 'difficult', not only for the method employed in the selection

process but also for the other face alignment methods. In other words this shows that the methods that we have examined have difficulties with the same images.

Second, we evaluate the consistency across different approaches, i.e., the consistency of 'difficult' samples found by different approaches. Thus, we calculate the pairwise consistency of $\mathcal{S}^i_{e_m}$ of those methods as shown in Fig. 5.5b. The resulting values are clearly much higher than the chance value of 0.22. In Fig. 5.5c we depict the 'optimal' case where the ground truth, that is the alignment error itself, is used to calculate the pairwise consistency. We observe that the consistency calculated by our selection process is very close to the one calculated based on the ground truth. We can further conclude that:

- the difficulty of samples is shared by the different methods that we have examined.
- the difficult samples selected by the mirror error show high consistency across different approaches.

5.4.2 Feedback on cascaded face alignment

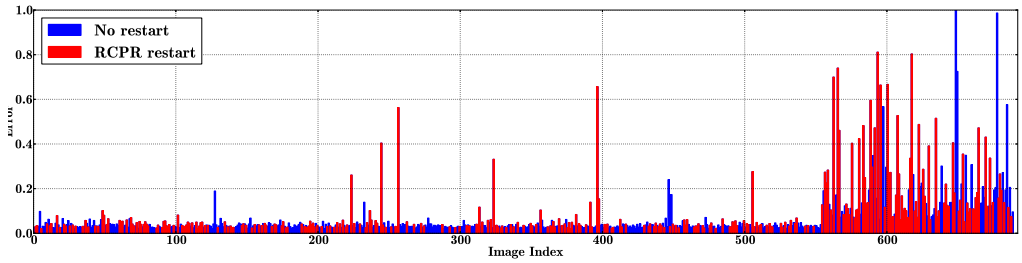
In recent years cascaded methods like SDM (Xiong and De la Torre, 2013), IFA (Asthana et al., 2014), CFAN (Zhang et al., 2014a) and RCPR (Burgos-Artizzu et al., 2013) have shown promising results in face alignment. Although they differ in terms of the regressor and the features that they use in each iteration they all follow the same strategy. The methods start from one or several initializations of the face shape, that are often calculated from the face bounding box, and then iteratively refine the estimation of the face shape by applying at each iteration a regressor that estimates the udpate of the shape. These methods are intrinsically sensitive to the initialization (Burgos-Artizzu et al., 2013; Zhang et al., 2014a) . As stated in (Xiong and De la Torre, 2014), only initializations that are in a range of the optimal shape can converge to the correct solution. To address this problem, (Cao et al., 2012) proposed to use several random initializations and give the final estimate as the median of the solutions to which they convergence. However, having several randomly generated initializations does not guarantee that the correct solution is reached. The 'smart restart' proposed in (Burgos-Artizzu et al., 2013) has improved the results to a certain degree. The scheme starts from different initializations and apply only 10% of the cascade. Then, the variance between the predictions is checked. If the variance is below a certain threshold, the remaining 90% of the cascade is applied as usual. Otherwise the process is restarted with a different set of initializations.

Here, we propose to use the mirror error as a feedback to close this *open* cascaded system. More specifically, for a given test image we first create its mirror image. Then we apply the RCPR model on the original test image and the mirror image and calculate the

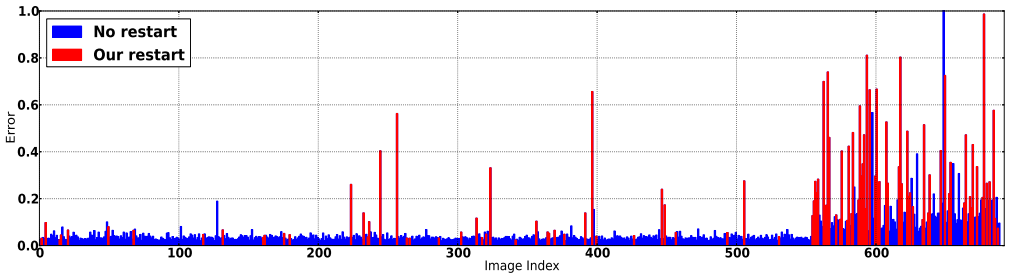
mirror error. If the mirror error is above a threshold we restart the process using different initializations, otherwise we keep the detection results. This procedure can be applied until the mirror error is below a threshold, or until a maximum number of iterations M is reached. In contrast to the original RCPR method that keeps only the results from the last set of initializations, we keep the one that has the smallest mirror error. This makes sense since new random initializations do not necessarily lead to better results than past initializations.

First we evaluate the effectiveness of our feedback scheme. Ideally, the restart will be initiated only when the current initialization is unable to lead to a *good* solution. Treating it as a two class classification problem we report results using a precision-recall based evaluation. A face alignment is considered to belong to the 'good' class if the mean alignment error is below 10% of the inter-ocular distance, otherwise, it is considered to belong to the 'bad' class - in the latter case a re-start is needed. The precision is the number of samples classified correctly as belonging to the 'bad' (positive) class divided by the total number of samples that are classified as belonging to the 'bad' class. Recall in this context is defined as the number of true positives divided by the total number of samples that belong to the bad class. For a fair comparison, we adjust our threshold on the mirror error (i.e. the threshold above which we restart the cascade with a different initialization) to get similar recall as the RCPR with smart re-start (Burgos-Artizzu et al., 2013) gets using its default parameters. We note that our parameter can also be optimized by cross validation for better performance. As can be seen in Fig. 5.6, at a similar recall level, our proposed scheme has significantly higher precision (0.65 vs. 0.25) than that of RCPR 'smart re-start', this verifies that our method is more effective in selecting samples for which restarting initializations are needed.

Second, we evaluate the improvement in the face alignment that we obtain using our proposed feedback scheme. We compare to 1) RCPR without restart (RCPR-O), 2) RCPR with the smart restart of (Burgos-Artizzu et al., 2013) (RCPR-S) and 3) other state of the art methods. We create two versions of our method. The first version, RCPR-F1, uses 5 initializations and at most two restarts - this allows direct comparison to the baseline method that uses the same number of initializations and restarts. The second version, RCPR-F2, uses 10 initializations and at most 4 times of restarts - this version produces better results and still has good runtime performance. We compare to SDM (Xiong and De la Torre, 2013), IFA (Asthana et al., 2014), GN-DPM (Tzimiropoulos and Pantic, 2014) and CFAN (Zhang et al., 2014a) - all of those have publicly available software and report good results. The results of the comparison is shown in Table 5.1. We compare the normalized alignment error of the common 49 inner facial landmarks for all of these methods and the 68 facial landmarks whenever this is possible. On the challenging 300W test set, with our proposed feedback scheme, the RCPR method has the best performance compared to not only the



(a) Original RCPR restart scheme. Presion=0.25, Recall = 0.63.



(b) Our restart scheme. Precision = 0.65, Recall = 0.63.

Fig. 5.6 Restart scheme of our method vs. RCPR (Burgos-Artizzu et al., 2013) (best viewed in color).

Methods	*RCPR-F2*	*RCPR-F1*	RCPR-S	RCPR-O	SDM	IFA	GN-DPM	CFAN
49P	**5.35**	6.07	6.59	7.14	7.12	8.31	12.42	7.24
68P	**6.25**	7.11	7.42	7.73	-	-	-	7.72

Table 5.1 49/68 facial landmark mean error comparison .

original version of RCPR but also to all the other methods. Although good performance is obtained on the face alignment problem, we emphasize that the main focus of this work is to bring attention to the mirroability of object localisation models.

5.5 Comprehensive comparison

Based on the previous setting, we compare all the holistic algorithms we have proposed in these thesis, including the Cascaded Forest (CasF) from Section 3.2, the RSSDM in Section 3.3, RF + RCPR model from Section 4.2, RCPR with mirrorability re-start (Mirror-RCPR in this Chapter) and Cascaded Forest with similar mirrorability re-start (Mirror-CasF). Except the RF model that is trained on AFLW in Section 2.3, the cascaded models are all trained on the training set of the 300W with the same augmentation setting. The models are tested on the 689 test images based on the split we discussed before. The performance is shown in Fig. 5.7. As can be seen, the Mirror-CasF, Mirror-RCPR and RF + RCPR perform quite

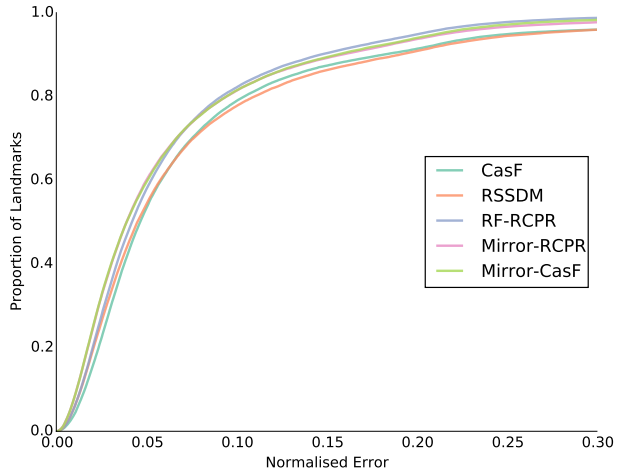


Fig. 5.7 Comprehensive comparison of our proposed holistic algorithms on 300w dataset.

similarly, that are better than RSSDM and the original CasF. RF + RCPR achieves the top performance. In our experiments we find those algorithms fail on some common images that are under extreme head variations or with very heavy occlusions. On the contrary, the training data has very few such samples. Though Mirror-CasF, Mirror-RCPR and RF + RCPR are modelled in different ways, they all achieve nearly saturation performance on the benchmark dataset. We note that the RF + RCPR model estimates occlusion explicitly by an additional regression forest model which further boost the performance slightly–as there are only a few samples are with heavy occlusion in the test set. In terms of run-time performance, CasF and RCPR based methods are much faster and process one face alignment in millisecond on a standard desktop. RSSDM and RF-RCPR are relatively slower, process 20 to 30 face images per second on average. In terms of memory cost at testing time, forests based methods (CasF and Mirror-CasF) and RSSDM are smaller than RCPR based methods. These pros and cons should be taken into account in practical application.

5.6 Summary and Discussion

In this chapter, we have investigated how state of the art facial landmark localisation methods behave on mirror images in comparison to how they behave on the original ones. All of the methods that we have evaluated struggle to get mirror symmetric results despite the fact that they were trained with datasets that were augmented with the mirror images.

In order to qualitatively analyse their behaviour, we introduced the concept of mirrorability and defined a measure called the mirror error. Our analysis led to some interest-

ing findings in mirrorability, among which a high correlation between the mirror error and ground truth error. Further, since the ground truth is not needed to calculate the mirror error, we show two applications, namely difficult samples selection and cascaded face alignment feedback that aids a re-initialization scheme. We believe there are many other potential applications in particular in Active Learning.

We also have carried out experiments and found the same holds for human pose estimation and for a variety of methods.

The findings of this work raise several interesting questions. Why some methods have shown better performance in terms of absolute mirror error, for example SDM is smaller and RCPR is bigger? Can the design of algorithms with low mirrorability error lead to algorithms with good overall performance? We believe these are all interesting research problems for future work.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we present a set of methods, local-based and holistic-based, for the problem of face alignment in real-world scenarios where acquisition of the face images cannot be controlled.

In chapter 2, we developed local based regression forest methods for face alignment. We use privileged information such as head pose, that is only available during the training time, to build better trees and learn conditional models, which can be generally regarded as a better local expert detector. We also learn shape constraints within the forest. We propose a regression forest vote fine-tuning scheme. It sieves invalid votes that are inconsistent to latent global variables like face center location and aggregates votes when necessary to introduce stronger shape constraints. The proposed methods demonstrated better performance than the standard regression forests on datasets collected from both constrained (BioID) and unconstrained conditions (LFW, AFLW and LFPW). We also evaluated the generality by applying it on a relevant problem, car alignment (CMU-Car dataset) and get promising performance.

In chapter 3, we proposed holistic based methods including regression forest based cascaded methods, and random subspace supervised descend method. We release a demo system with source code for future comparison. The demo system works efficiently on face alignment for ordinary and sketch images in the wild.

In chapter 4, we developed methods specifically for robust face alignment under heavy occlusion. One is structured semi-supervised forest framework for simultaneously face alignment and occlusion reasoning, using a subset of face images with face mask annotation. The other one models face occlusion in an unsupervised way by introducing the concept of regional predictive power. Both of these two methods do face alignment and

face mask prediction simultaneously. Thus the output consists of the locations of the facial landmarks and the dense labelling of face image that indicates the probability of each pixel belong to the face or not. To the best of our knowledge, this is the first system that combines face alignment and dense face mask prediction. We believe the explicit occlusion prediction will be useful for subsequent face analysis. In order to evaluate the performance of the face mask prediction, we extended the COFW dataset with pixel level face mask labelling and form the COFWM dataset, which is publicly available for future research use.

In chapter 5 we introduced the concept of mirrorability and the measure mirror error to evaluate the ability of a model/algorithm to preserve the mirror symmetry when applied on an image and its mirror image. We show that the mirror error is highly correlated to alignment error for several methods in face alignment. We demonstrated it can be used for selecting difficult samples and serving as a feedback for cascaded face alignment method.

In summary, we have focused on solving the problem of face alignment of images collected in unconstrained environments and proposed several local based and holistic based methods. On the benchmark datasets, the holistic methods perform better than the local based methods and most of the current researches also focus on developing holistic methods. However, as we discussed in chapter 4. The holistic methods are intrinsically quite sensitive to initialization, that is usually generated from the result of the face detection. On the contrary, face detection shift has much less impact on the local based methods we proposed in this thesis. Moreover, since the holistic methods regress the face shape as a whole, they also fail in a holistic way while the detection results of the local based methods are less correlated. In terms of computational complexity, holistic based methods hold advantages over the local based methods. This is more obvious when the number of the landmarks is large. The computation of local based methods, for local detection and global optimization often increases exponentially, while that of the holistic methods usually increases very tiny.

6.2 Future Work

Face alignment has made significant progress in the past years. Before 2011, most of the methods were working on face images collected in the laboratory like the BioID dataset. In recent years, several new datasets have been created with images collected from the Internet such as LFPW, LFW, AFPW and COFW. The research community made impressive improvements and report very good performance on these datasets. The methods proposed in this thesis achieve better or comparable performance to the state of the art methods.

However, for many real applications, robustness, accuracy, efficiency are still not sufficient. For example, we observe that most of the methods do work well on images from

the iBug-sub set of the 300W dataset where the images are significantly more challenging in terms of head pose variations and occlusion. Moreover, it is also challenging to detect the faces in those images, which leads to more difficult initialization. In the following we present possible future work.

Unified face detection and face alignment. For most of the face alignment evaluation dataset, face detection results are provided using the same detector or manually labelled in the same definition. These two problems are quite dependent. (Pedersoli et al., 2014; Ren et al., 2014; Zhu and Ramanan, 2012) have exploited the possibility of addressing them together. However, face detection itself remains difficult when face is under heavy occlusion. Also, most of the current face alignment system, particularly the well performing holistic cascaded systems, are very sensitive to face detection changes. Thus when a face detection at testing time varies from that is used during test time, the performance of a system drops sharply. In order to develop a robust system, more attention should be brought when linking these two steps.

Task-oriented face alignment. Currently, we usually separate the face alignment task from the face analysis pipeline and attempt to get the best performance on the face alignment benchmarks. There are several questions that need to be addressed. For example, how many facial landmarks are required? Are the landmarks along the face contour necessary? How accurate is sufficient for a specific application? How does erroneous face alignment influence face recognition or some other applications? We believe these are all interesting questions to answer in future work. For instance, in facial expression recognition research, some work even requires sub-pixel level registration, of which even the ground truth of facial landmark locations is very difficult to obtain.

Face alignment validation system. Like most of other computer systems, most face alignment methods do not validate the reliability of the results. As a median step in the DAR pipeline discussed in the introduction chapter, erroneous result in this step will lead to failures in the next step, thus a failure alert system in very necessary in practice. As we demonstrated in chapter 5, the mirror error can be served as an efficient and reliable way of failure checking. There are much open space in developing failure check systems based on this concept.

Extension to relevant problems. Face alignment has made rapid progress in recent years and achieved good accuracy, but some relevant problems like human pose estimation and bird part localisation remain very challenging. The development in face alignment are likely to be adapted and extended for problems in the similar domains.

Mirrorability in computer vision. In chapter 5 we have introduced the concept of mirrorability and evaluated it on two representative problems. There are many interesting

questions that remain unsolved. For example, what is the impact of mirrorability if the data is mirrored? what is the impact when the models are mirrored? How mirror error exhibit in different types of models? We believe these problems are all interesting and worthy further investigation.

References

B Amberg and T Vetter. Optimal landmark detection using shape models and branch and bound. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 455–462, 2011.

Brian Amberg, Andrew Blake, and Thomas Vetter. On compositional image alignment, with an application to active appearance models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1714–1721. IEEE, 2009.

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5): 898–916, 2011.

Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3444–3451, 2013.

Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1859–1866, 2014.

Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Continuous conditional neural fields for structured regression. In *Proc. Eur. Conf. Comput. Vis.*, pages 593–608. Springer, 2014.

P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 545–552, 2011.

Betaface. http://www.betaface.com/.

W. W. Bledsoe. The Model Method in Facial Recognition. *Technical Report PRI 15*, page 47, 1964.

W. W. Bledsoe and H. Chan. A Man-Machine Facial Recognition System-Some Preliminary Results. *Technical Report PRI 19A*, 1965.

Vishnu Naresh Boddeti, Takeo Kanade, and BVK Vijaya Kumar. Correlation filters for object alignment. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2291–2298, 2013.

A Bosch, A Zisserman, and X Muoz. Image classification using random forests and ferns. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–8, 2007.

L. Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1513–1520, 2013.

Xudong Cao, Y. Wei, F. Wen, and Jian Sun. Face alignment by explicit shape regression. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 177–190. Springer, 2012.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. on Intell. Syst. Technol.*, 2:27:1–27:27, 2011. available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Proc. Int'l Conf. Image Processing*, pages 34–37, 2001.

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.

T.F. Cootes, M. C.Lindner Ionita, and Sauer P. Robust and accurate shape model fitting using random forest regression voting. In *Proc. Eur. Conf. Comput. Vis.*, pages 278–291. Springer, 2012.

Timothy F Cootes and Christopher J Taylor. Combining point distribution models with shape models based on finite element analysis. *Image Vision Comput.*, 13(5):403–409, 1995.

S. Coşar and M. Çetin. A graphical model based solution to the facial feature point tracking problem. *Image Vision Comput.*, 29(5):335–350, 2011.

M Cox, J Nuevo-Chiquero, JM Saragih, and S Lucey. Csiro face analysis sdk. *Proc. IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, 2013.

A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pages 106–117, 2011a.

Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, (7): 81–227, 2011b.

D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *Proc. Brit. Mach. Vis. Conf.*, volume 2, page 6, 2006.

D Cristinacce and T Cootes. Boosted regression active shape models. In *Proc. Brit. Mach. Vis. Conf.*, volume 2, pages 880–889, 2007.

D Cristinacce and T Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, volume 1, pages 886–893, 2005.

M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2578–2585, 2012a.

M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2578–2585, 2012b. ISBN 978-1-4673-1228-8. doi: 10.1109/CVPR.2012.6247976.

Fernando De la Torre and Minh Hoai Nguyen. Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1–8. IEEE, 2008.

D F DeMenthon and L S Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vis.*, 15(1):123–141, 1995.

P Dollár, P Welinder, and P Perona. Cascaded pose regression. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1078–1085, 2010.

Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1841–1848. IEEE, 2013.

B. Efraty, C. Huang, SK Shah, and IA Kakadiaris. Facial landmark detection in uncontrolled conditions. In *Proc. Int'l Joint Conf. Biometrics*, pages 1–8, 2011.

Hazım Ekenel and Rainer Stiefelhagen. Why is facial occlusion a challenging problem? *Advances in Biometrics*, pages 299–308, 2009.

G Fanelli, J Gall, and L Van Gool. Real time head pose estimation with random regression forests. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 617–624, 2011.

Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 59(2):167–181, 2004.

John R Finnerty. Did internal transport, rather than directed locomotion, favor the evolution of bilateral symmetry in animals? *BioEssays*, 27(11):1174–1180, 2005.

J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2188–2202, 2011.

Juergen Gall and Victor Lempitsky. Class-specific hough forests for object detection. In *Decision Forests for Computer Vision and Medical Image Analysis*, pages 143–157. Springer, 2013.

Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Multi-channel correlation filters. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3072–3079, 2013.

Xinbo Gao, Nannan Wang, Dacheng Tao, and Xuelong Li. Face sketch–photo synthesis and retrieval using sparse representation. *IEEE Trans. Circuits Syst. Video Technol.*, 22(8): 1213–1226, 2012.

Yongsheng Gao, Maylor KH Leung, Siu Cheung Hui, and Mario W Tananda. Facial expression recognition from line-based caricatures. *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(3):407–412, 2003.

Robert Gens and Pedro Domingos. Deep symmetry networks. In *Advances in Neural Information Processing Systems*, pages 2537–2545, 2014.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, pages 3–42, 2006.

Golnaz Ghiasi and Charless C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1899–1906, 2014.

R Girshick, J Shotton, P Kohli, A Criminisi, and A Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 415–422, 2011.

Ben Glocker, Olivier Pauly, Ender Konukoglu, and Antonio Criminisi. Joint classification-regression forests for spatially structured multi-object segmentation. pages 870–881. Springer, 2012.

P. Grassberger. Entropy estimates from insufficient samplings. *Arxiv preprint physics/0307138*, 2003.

R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image Vision Comput.*, 23(12):1080–1093, 2005.

Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. volume 28, pages 807–813. Elsevier, 2010.

Tin Kam Ho. Random decision forests. In *Proc. 3rd Int'l Conf. Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.

Gary B Huang. *Weakly supervised learning for unconstrained face processing*. PhD thesis, UNIVERSITY OF MASSACHUSETTS AMHERST, 2012.

Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and CV Jawahar. Has my algorithm succeeded? an evaluator for human pose estimators. In *Proc. Eur. Conf. Comput. Vis.*, pages 114–128. Springer, 2012.

O Jesorsky, K Kirchberg, and R Frischholz. Robust face detection using the hausdorff distance. In *Proc. Int'l Conf. Audio and Video-based Biometric Person Authentication*, pages 90–95. Springer, 2001.

Xuhui Jia, Heng Yang, Angran Lin, Kwok-Ping Chan, and Ioannis Patras. Structured semi-supervised forest for facial landmarks localization with face mask reasoning. In *Proc. Brit. Mach. Vis. Conf.*, 2014.

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

A Kasinski, A Florek, and A Schmidt. The put face database. *Image Processing and Communications*, 13(3-4):59–64, 2008.

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1867–1874. IEEE, 2014.

E. M. Kleinberg. Stochastic discrimination. *Annals of Mathematics and Artificial Intelligence*, 1:207–239, 1990.

EM Kleinberg et al. An overtraining-resistant stochastic modeling method for pattern recognition. *The annals of statistics*, 24(6):2319–2349, 1996.

Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1568–1583, 2006.

P Kontschieder, S R Bulo, H Bischof, and M Pelillo. Structured class-labels in random forests for semantic image labelling. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2190–2197, 2011.

Peter Kontschieder, Samuel Rota Bulò, Michael Donoser, Marcello Pelillo, and Horst Bischof. Evolutionary hough games for coherent object detection. *Comput. Vis. Image Understand.*, pages 1149–1158, 2012.

Peter Kontschieder, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi. Geof: Geodesic forests for learning coupled predictors. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 65–72, 2013.

M. Kostinger, P. Wohlhart, P.M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, pages 2144–2151. IEEE, 2011.

Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. Enhanced pictorial structures for precise eye localization under incontrolled conditions. In *Proc. Eur. Conf. Comput. Vis.*, pages 1621–1628, 2012.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proc. Eur. Conf. Comput. Vis. Workshop on Statistical Learning in Computer Vision,*, pages 17–32. Springer, 2004.

Yan Li, Leon Gu, and Takeo Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(9):1860–1876, 2011.

L Liang, R Xiao, F Wen, and J Sun. Face alignment via component-based discriminative search. In *Proc. Eur. Conf. Comput. Vis.*, pages 72–85. Springer, 2008.

David G Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Int. Conf. Comput. Vis.*, volume 2, pages 1150–1157. Ieee, 1999.

David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *Proc. Eur. Conf. Comput. Vis.*, pages 508–521. Springer, 2006.

Simon Lucey, Rajitha Navarathna, Ahmed Bilal Ashraf, and Sridha Sridharan. Fourier lucas-kanade algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1383–1396, 2013.

B Martinez, M Valstar, X Binefa, and M Pantic. Local Evidence Aggregation for Regression Based Facial Point Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1149–1163, 2012.

Kieron Messer, Jiri Matas, Josef Kittler, Juergen Luettin, and Gilbert Maitre. Xm2vtsdb: The extended m2vts database. In *Proc. Int'l Conf. Audio and Video-based Biometric Person Authentication*, volume 964, pages 965–966, 1999.

Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *Proc. Eur. Conf. Comput. Vis.*, pages 504–513. Springer, 2008.

S Moore and R Bowden. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding*, 115(4):541–558, 2011.

Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. Int'l Conf. Computer Vision Theory and Applications*, pages 331–340, 2009.

S. Nowozin. Improved information gain estimates for decision tree induction. In *Int'l Conf. Machine Learning*, 2012.

Marco Pedersoli, Radu Timofte, Tinne Tuytelaars, and Luc Van Gool. Using a deformation field model for localizing faces and facial points under weak supervision. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3694–3701. IEEE, 2014.

J Ross Quinlan. C4. 5: Programming for machine learning. *Morgan Kauffmann*, 1993.

V Rapp, T Senechal, K Bailly, and L Prevost. Multiple kernel learning svm and statistical validation for facial landmark detection. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit.*, pages 265–271, 2011.

Nima Razavi, Juergen Gall, Pushmeet Kohli, and Luc Van Gool. Latent hough transform for object detection. In *Proc. Eur. Conf. Comput. Vis.*, pages 312–325. Springer, 2012.

Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1685–1692, 2014.

Myung-Cheol Roh, Takaharu Oguri, and Takeo Kanade. Face alignment robust to occlusion. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit.*, pages 239–244, 2011.

Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 38–44, 1998.

C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 896–903, 2013a.

Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 896–903. IEEE, 2013b.

Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 397–403, 2013c.

Jason Saragih and Roland Goecke. A nonlinear discriminative approach to aam fitting. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–8, 2007.

Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable face fitting with soft correspondence constraints. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit.*, pages 1–8. IEEE, 2008.

Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. Comput. Vis.*, 91(2):200–215, 2011.

Walter J Scheirer, Anderson Rocha, Ross J Micheals, and Terrance E Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1689–1695, 2011.

Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. Detecting and aligning faces by image retrieval. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3460–3467, 2013.

Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1–8. IEEE, 2008.

Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

Brandon M Smith and Li Zhang. Collaborative facial landmark localization for transferring annotations across datasets. In *Proc. Eur. Conf. Comput. Vis.*, pages 78–93. Springer, 2014.

Brandon M Smith, Jonathan Brandt, Zhe Lin, and Li Zhang. Nonparametric context modeling of local appearance for pose-and expression-robust facial landmark localization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1741–1748, 2014.

Peter D Sozou, Timothy F Cootes, Christopher J Taylor, EC Di Mauro, and Andreas Lanitis. Non-linear point distribution modelling using a multi-layer perceptron. *Image Vision Comput.*, 15(6):457–463, 1997.

M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3394–3401, 2012.

Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3476–3483, 2013.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3224–3231, 2013.

Xiaoou Tang and Xiaogang Wang. Face sketch recognition. *IEEE Trans. Circuits Syst. Video Technol.*, 14(1):50–57, 2004.

Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *Proc. Eur. Conf. Comput. Vis.*, pages 41–54. Springer, 2012.

Georgios Tzimiropoulos and Maja Pantic. Optimization problems for fast aam fitting in-the-wild. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 593–600, 2013a.

Georgios Tzimiropoulos and Maja Pantic. Optimization problems for fast aam fitting in-the-wild. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 593–600. IEEE, 2013b.

Georgios Tzimiropoulos and Maja Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1851–1858, 2014.

Georgios Tzimiropoulos, J Medina, S Zafeiriou, and Maja Pantic. Active orientation models for face alignment in-the-wild. *IEEE Trans. Inf. Forensics Security*, 9(12):2024–2034, Dec 2014.

M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2729–2736, 2010.

V Vapnik and A Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

P. Viola and M.J. Jones. Robust real-time face detection. *Int. J. Comput. Vis.*, 57(2):137–154, 2004.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages I–511, 2001.

D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, pages 1692–1698, 2005.

Nannan Wang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Jie Li. A comprehensive survey to face hallucination. *Int. J. Comput. Vis.*, 106(1):9–30, 2014.

Peng Wang, Qiang Ji, and James L Wayman. Modeling and predicting face recognition system performance based on analysis of similarity scores. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(4):665–670, 2007.

Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):1955–1967, 2009.

Laurenz Wiskott, J-M Fellous, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):775–779, 1997.

Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 532–539, 2013.

Xuehan Xiong and Fernando De la Torre. Supervised descent method for solving nonlinear least squares problems in computer vision. *arXiv preprint arXiv:1405.0601*, 2014.

Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Learn to combine multiple hypotheses for accurate face alignment. In *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, pages 392–396, 2013.

Fei Yang, Junzhou Huang, and Dimitris Metaxas. Sparse shape registration for occluded facial feature localization. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit.*, pages 272–277, 2011.

H Yang and I Patras. Privileged information-based conditional structured output regression forest for facial point detection. *IEEE Trans. Circuits Syst. Video Technol.*, 2015.

Heng Yang and Ioannis Patras. Face parts localization using structured-output regression forests. In *Assian Conf. Computer Vision*, pages 667–679. Springer, 2012.

Heng Yang and Ioannis Patras. Privileged information-based conditional regression forests for facial feature detection. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit.*, 2013a.

Heng Yang and Ioannis Patras. Sieving regression forests votes for facial feature detection in the wild. In *Proc. Int'l Conf. Computer Vision*. IEEE, 2013b.

Heng Yang and Ioannis Patras. Fine-tuning regression forests votes for object alignment in the wild. *IEEE Trans. Image Processing*, 2014.

Heng Yang, Changqing Zou, and Ioannis Patras. Cascade of forests for face alignment. *IET Computer Vision*, 2014.

Heng Yang, Xuming He, Xuhui Jia, and Ioannis Patras. Robust face alignment under occlusion via regional predictive power estimation. *IEEE Trans. Image Processing*, 2015a.

Heng Yang, Xuhui Jia, Ioannis Patras, and K.P. Chan. Random subspace supervised descent method in computer vision. *IEEE Signal Processing Letters*, 2015b.

Jun Yu, Meng Wang, and Dacheng Tao. Semisupervised multiview distance metric learning for cartoon synthesis. *IEEE Trans. Image Processing*, 21(11):4636–4648, 2012.

Xiang Yu, Junzhou Huang, Shaoting Zhang, Wang Yan, and Dimitris N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1944–1951, 2013a.

Xiang Yu, Fei Yang, Junzhou Huang, and DN Metaxas. Explicit occlusion detection based deformable fitting for facial landmark localization. In *Proc. IEEE Int'l Conf. on Autom. Face Gesture Recognit. Workshop*, pages 1–6, 2013b.

Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Proc. Eur. Conf. Comput. Vis.*, pages 1–16. Springer, 2014a.

Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *Proc. Eur. Conf. Comput. Vis.*, pages 1–16. Springer, 2014b.

Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. Predicting failures of vision systems. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3566–3573, 2014c.

Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 513–520, 2011.

Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Proc. Eur. Conf. Comput. Vis.*, pages 94–108. Springer, 2014d.

Feng Zhou, Jonathan Brandt, and Zhe Lin. Exemplar-based graph matching for robust facial landmark localization. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1025–1032, 2013.

Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Transferring landmark annotations for cross-dataset face alignment. *arXiv preprint arXiv:1409.0602*, 2014.

X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 2879–2886, 2012. http://www.ics.uci.edu/~xzhu/face/.

Appendix A

Sketch Face Alignment

In this appendix, we present a cascaded face alignment method for both RGB images and sketch images. We first propose a scheme to synthesize face sketches from face photos based on random-forests edge detection and local face region enhancement. Then we jointly train a Cascaded Pose Regression based method for face alignment for both face photos and sketches. We build an evaluation dataset, called Face Sketches in the Wild (**FSW**), with 450 face sketch images collected from the Internet and with the manual annotation of 68 facial landmark locations on each face sketch. The proposed multi-modality facial landmark localization method shows competitive performance on both face sketch images and face photo images.

A.1 Problem definition

Sketches are frequently used as a means of visual representation of an individual's face. Such representation has been applied for digital entertainment like cartoon synthesis (Wang et al., 2014; Yu et al., 2012), facial expression recognition (Gao et al., 2003), face retrieval (Gao et al., 2012) and face recognition in law enforcement (Wang and Tang, 2009; Zhang et al., 2011). In the latter case, the photo of a suspect is not available and the face sketch is drawn based on the information collected from the witnesses. Taking the sketch retrieval and photo-to-sketch face recognition as an example, the challenge of using sketch representation mainly lies in the modality difference between the sketch and the photo. Several approaches (Gao et al., 2012; Tang and Wang, 2004; Wang et al., 2014; Wang and Tang, 2009; Zhang et al., 2011) focus on bridging the gap of the two modalities. Similar to photo-to-photo face recognition, it is crucial to align the face sketch first into a canonical pose, where the face pose is always represented by a set of facial landmarks.

In recent years, facial landmarks localization (or face alignment) has made a significant

progress on face images in the wild, using the holistic pose regression methods (Burgos-Artizzu et al., 2013; Cao et al., 2012; Tzimiropoulos and Pantic, 2013a; Xiong and De la Torre, 2013; Yang and Patras, 2012, 2013a), or local based methods (Asthana et al., 2013; Smith et al., 2014; Yang and Patras, 2013b; Zhou et al., 2013). However, due to the modality difference, the performance drops significantly on face sketches. In this work we address this problem, in order to make applications like sketch-to-photo face recognition and face sketch retrieve more practical in real world.

Only a few face sketch datasets are currently available. In most of them, like the CUFSF (Wang and Tang, 2009) and CUFS (Zhang et al., 2011), the sketches are drawn by artists based on original face photos. Some sketches, like that in CUFSF are with shape exaggeration. These sketch images are not as challenging as those from the real world in two aspects: first, the original photos which the sketches synthesized from were taken from constrained frontal poses (Wang and Tang, 2009; Zhang et al., 2011) while the sketches in real world might be in arbitrary poses; second, the high quality in terms of facial details of the sketches in those datasets is difficult to be obtained in real world application. Due to these limitations of datasets, it is difficult to train and to evaluate a general alignment model for face sketches. In order to deal with this, we propose to train a model for multi-modality facial landmark localization, by making full use of the publicly available face photo datasets collected in the wild with landmarks annotations. More specifically, we automatically generate sketches from images in those datasets by fusing local region enhancement and edge detection using structured random forests. We then train a Cascaded Pose Regression based on both the face photos and face sketches using the ground truth landmark annotation. The proposed method is illustrated in Fig. A.1.

In order to evaluate the performance of the proposed method, we collect face sketches from the Internet and create the Face Sketches in the Wild (**FSW**[1]) dataset. We compare our method with the current state-of-the art facial landmarks localization methods. We achieve almost the same results to the Robust Cascaded Pose Regression (Burgos-Artizzu et al., 2013) method trained on RGB images on the face photo dataset and the best performance on the FSW dataset.

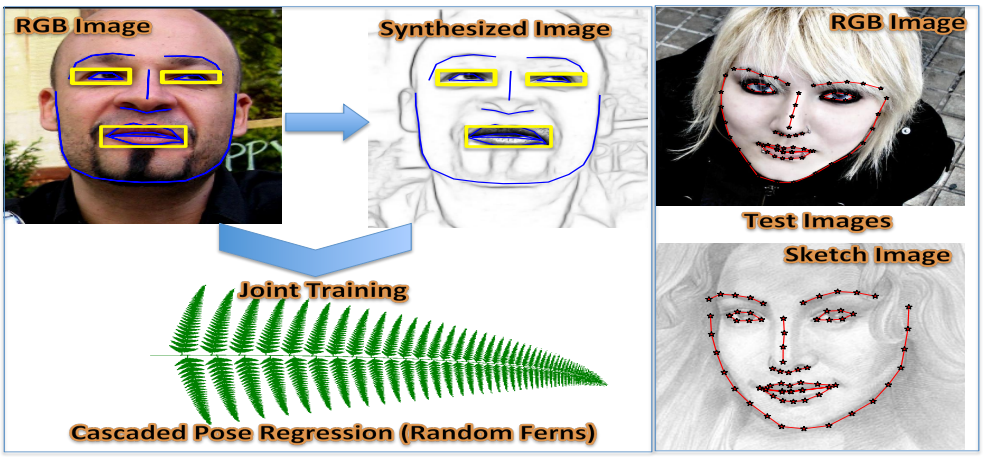


Fig. A.1 Our approach trains a Cascaded Pose Regression model based on RGB face images and their synthesis (left), then estimates the facial landmarks locations in both face photos and face sketch images (right).

Fig. A.2 An example image of face sketch synthesis. From left to right are the original RGB image, edge detection by (Dollár and Zitnick, 2013) and our synthesized sketch image. In the synthesized image the eye regions and mouth region are enhanced and fused with the edge detection.

A.2 Method

Face sketch synthesis

Most of the current face sketch synthesis approaches follow a supervised learning route, for instance the Markov Random Fields (MRF) in (Wang and Tang, 2009) require a large number of *ground truth* face sketches that are often drawn by artists. It is quite expensive to acquire such training samples since face alignment model training often demands thousands of training instances. Moreover, using only drawings made by artists limits the diversity of the face sketches. In applications such as face sketch retrieval, the sketch might be drawn by non-experts, that have key differences from the drawing drawn by artists.

As opposed to a supervised learning synthesis, our face sketch synthesis scheme is based on fast edge detection using structured forests (Dollár and Zitnick, 2013). Note that though this edge detection method is learning based, it is trained for general edge detection. We define it as a non-learning based method because it is not necessary or required to have face sketches at the training phrase. More specifically, we assume we are given a set of face photos $\mathcal{I} = \{I\}$, where for each I we have its annotation of facial landmark locations. We denote the structured forests based edge detector by $\mathcal{F}$, thus given an image photo I, the edge detection result of I is:

$$I^e = \mathcal{F}(I). \tag{A.1}$$

I^e, as shown in Fig. A.2 contains global shape information like the contour of the face but lacks details in local regions such as the eye shapes and mouth lips. However, the eye and mouth regions are important features on the face thus they are often depicted in detail in sketch images. In order to synthesize the sketch with more details in these regions, we use their enhanced gray scale images. More specifically, we extract the rectangles around the two eye regions and the mouth region, based on the ground truth locations of their boundary landmarks. After converting the RGB image patches into gray scale, we further apply histogram equalization to increase the global contrast. Then the synthesized sketch is represented by:

$$I^s = I^e \oplus (I_{leye} \cup I_{reye} \cup I_{mouth}) \tag{A.2}$$

where I_{leye}, I_{reye} and I_{mouth} are the enhanced gray scale images from left eye region, right eye region and mouth region, respectively. The operator $\oplus$ works in a way of putting the layer of the right side on top of the layer of the left side, i.e. to replace the content of I^s at the corresponding pixels with the enhanced gray scale images. An example image is shown in Fig. A.2. Although this procedure is very simple, the result looks very similar to sketch

[1] https://sites.google.com/site/yanghengcv

images. Its effectiveness in improving the landmark localization performance on sketch images will be demonstrated in the experimental section.

A.2.1 Joint training of cascaded pose regression

We use the Cascaded Pose Regression (CPR) (Dollár et al., 2010) framework in this work given its efficiency and accurate performance for estimating face landmark locations (Burgos-Artizzu et al., 2013; Cao et al., 2012). We follow the main steps of CPR evaluation procedure. A CPR consists of a cascade of T regressors $R^{1\ldots T}$. An estimation of a shape starts from a initial guess S^0, and progressively refine the estimation by an update in each iteration, until the final stage of regression is applied. As demonstrated in Algorithm 4, given

Algorithm 4 Cascaded Pose Regression

Input: Image I, initial pose S^0, regressors $R^{1\ldots T}$
Output: Estimated pose S^T

1: **for** t=1 to T **do**
2: $f^t = h^t(I, S^{t-1})$ ▷ Shape-indexed features
3: $\Delta S = R^t(f^t)$ ▷ Apply regressor R^t
4: $S^t = S^{t-1} + \Delta S$ ▷ update pose
5: **end for**

the estimation of pose in the previous iteration S^{t-1}, image feature for the t-th iteration are calculated as $f^t = h^t(I, S^{t-1})$. Based on the feature f^t and the regressor R^t, an update ΔS is calculated, once is added on the previous estimation of the pose. Similar to (Cao et al., 2012) and (Burgos-Artizzu et al., 2013), we use two stages of regression, i.e. at each iteration, multiple regressors are utilized and they share the same pose for feature calculation that is from the previous iteration. We also use the random fern as the primitive regressor and follow their training scheme that directly minimizes the alignment error. We use the interpolated shape-indexed features proposed in (Burgos-Artizzu et al., 2013). The latter uses a reference location between the locations of two landmarks thus is more robust against large pose variations and shape deformations.

As discussed before, we assume we have a dataset with face photo images and their facial landmarks annotation $\{(I_i, \hat{S}_i)\}_{i=1}^N$, where $\hat{S}_i$ is the vector of ground truth landmark locations. For each face photo I_i, we will generate a sketch synthesis as discussed in Eq. A.2. Thus we have an additional set of training samples $\{(I_i^s, \hat{S}_i)\}_{i=1}^N$, based on the assumption that the synthesized face sketch image shares the same facial landmark annotation with the face photo. Similar to (Burgos-Artizzu et al., 2013; Cao et al., 2012; Dollár et al., 2010), we augment the training samples by initializing them with several random poses from other

training samples. Like (Cao et al., 2012), each regressor is learnt by explicitly minimizing the sum of alignment errors. We adapt it by putting different weight on the error of sketch images and face photos, that is,

$$R^t = \arg\min_R (\alpha E_t(R) + (1-\alpha) E_t^s(R)) \tag{A.3}$$

where $E_t(R) = \sum_{i=1}^{N} ||\hat{S}_i - R(I_i, S_i^{t-1})||$ is the sum of errors calculated over the face photo samples and $E_t^s(R) = \sum_{i=1}^{N} ||\hat{S}_i - R(I_i^s, S_i^{s,t-1})||$ is the sum of errors calculated over the sketch samples. S_i^{t-1} is the shape of the i-th face photo sample estimated by the $t-1$ iteration and $S_i^{s,t-1}$ is that for the face sketch sample. By setting the values of α, we can adjust the relatively importance of face photos and face sketch images at the training stage. We note that, this parameter is not used during the testing stage once the regressor R^t is found. In this way, we can train the cascade of the regressors jointly for both face photos and face sketch images and the testing procedure is as described in Algorithm 4.

A.3 Evaluation

A.3.1 Dataset and implementation details

We train our model using the training images of HELEN, a dataset that is widely used for evaluating facial landmarks localization in the wild. HELEN consists of 2510 training images and 330 test images, that are collected from the Internet, from search engine results or from Flickr. Most of those images exhibit a very large variability in pose, lighting, expression as well as general imaging conditions. Many images exhibit partial occlusions that are caused by head pose, objects (e.g., glasses, scarf, food), body parts (hair, hands) and shadows. We use the facial landmark annotations provided by the iBug challenge (Sagonas et al., 2013c) for the following reasons: 1) most of the recent methods in facial landmark localization use the 68 facial landmark mark-up from Multi-PIE (Gross et al., 2010); 2) it is a good benchmark and makes future comparisons more direct. We use the 2510 training images to build our model.

The currently available face sketch datasets like CUFSF and CUFS are drawn by artists based on face images taken in very constrained environments and they exhibit very limited variability in terms of head poses, facial expressions and occlusions. Therefore, we produced a new and significantly more challenging dataset for evaluation, which we call Face Sketches in the Wild (FSW). We collect face sketch images from the Internet by searching using Google and Bing. The dataset is designed to present face sketches in real-world conditions. The sketches exhibit large head pose changes, resolution variability, occlusions and

more importantly, different sketch styles. Some example images are shown in Fig. A.3. We finally got 450 images for evaluation by excluding some non-face sketches such as the exaggerated cartoon face sketches. One face is detected in each sketch image by Viola-Jones face detector, followed by manual checking. Then we manually annotate the locations of 68 facial landmarks, with the mark-up used in Multi-PIE (Gross et al., 2010) and (Sagonas et al., 2013c). Note that we only use these images for evaluation but not for building the model.

Our implementation of the proposed method is based on the Robust Cascaded Pose Regression (RCPR) code provided by (Burgos-Artizzu et al., 2013). We use their default parameter setting, i.e., 50 boosted ferns at each iteration, 100 iterations in total, the number of features $F = 400$, depth 5 random ferns. When training the baseline method, the data augmentation factor is 20, i.e. 20 random initializations are used for each training sample. For our joint training, we set the data augmentation factor to 10 for a fair comparison since we double the number of traning examples by using the synthesized face sketches. We set the parameters $\alpha = 0.4$ in Eq. A.3 by cross validation. We re-wrote the code using C++ on a standard 3.30GHz CPU machine in order to get faster performance. An online demo is available on the FSW dataset web page where the user can upload images for testing. The face detection is carried out by OpenCV Viola-Jones face detector.

For better comparison, we also consider some other methods that are able to detect both inner landmarks and contour landmarks using the same mark-up of Multi-PIE. We consider two recent representative local based methods: the Discriminative Response Map Fitting (**DRMF**) in (Asthana et al., 2013) and the Optimized Part Mixtures model (**OPM**) in (Yu et al., 2013a). For DRMF we run its model with given face detections. For OPM, which combines face detection and landmarks localization, we manually remove the false face detections when calculating the errors. This actually favours it since the face detection failure cases are often challenging images.

We report the error, i.e., the Euclidean distance between the ground truth location and the estimation, as a fraction of the inter-ocular distance, similar to (Burgos-Artizzu et al., 2013; Cao et al., 2012).

A.3.2 Results on FSW

First we evaluate the performance of facial landmarks localization in sketch images, since this is the main aim of the proposed method, on the FSW dataset. We benchmark our method on the RCPR framework with the interpolated indexed feature in (Burgos-Artizzu et al., 2013). We do not use the full version since its training requires landmark visibility annotation. We do not use the re-start scheme of RCPR for a fair comparison since the

re-start might vary from one to another and is also time-consuming. Different versions of such RCPR are trained including 1) **RCPR-RGB**, trained only on RGB face photo images (in practice, the RGB images are converted to gray scale images for model training); 2) **RCPR-RGB+Edge**, jointly trained on RGB face photo images and the corresponding face edge images detected by (Dollár and Zitnick, 2013); 3) **RCPR-Synthesis** trained on the synthesized images only; 4) **RCPR-RGB+Synthesis**, jointly trained on the RGB images and the corresponding synthesized images. We also compare to other facial landmarks localization method, that are trained for face landmarks localization for face photo images.

We report the average landmark-wise error of all the 68 facial landmarks of the test images of FSW, shown in Fig. A.4. On this challenging dataset, the proposed method, RCPR-RGB+Synthesis significantly outperforms the others, both variations of the RCPR and the two local based models. The model learned using only the synthesized images for training (RCPR-Synthesis) has the worst performance among all RCPR variations. When comparing the results of RCPR-RGB+Edge to RCPR-RGB, we can observe the improvement for the landmarks along the contour but the performance drops for inner landmarks. This is very likely because the edge images captures very similar information to the face sketches along the contour but not the detail of the face inner parts. The local based methods, particularly the OMP, that were trained on RGB images, do not work well on the face sketch images, due to the change of the modality. The superior performance of RCPR-RGB+Synthesis over both the RCPR-RGB and RCPR-RGB+Edge validates the effectiveness of our proposed joint training scheme by using the RGB and synthesized images. It is worthy noting that, the improvement on the contour landmarks, which are generally regarded as more difficult parts, is more significant. We visualize the individual landmark error levels in the last image of Fig. A.3, from which we can observe the high localization accuracy of most of the facial landmarks.

A.3.3 Results on LFPW test images

We also evaluated the generality of the proposed method by evaluating the facial landmarks localization accuracy on RGB images. We report the performance on the LFPW, a test set which is wildly used for evaluating facial landmarks localization in the wild (Belhumeur et al., 2011; Burgos-Artizzu et al., 2013; Cao et al., 2012; Xiong and De la Torre, 2013; Zhou et al., 2013). The image in LFPW dataset which has much lower resolution than the HELEN dataset. The experiment is set in this way in a scenario the methods are trained on datasets different from the ones on which they are tested for fair comparison. The two local based methods, DRMF and OMP are trained on the Multi-PIE dataset while our RCPR variants are trained on the HELEN training images, all are tested on LFPW. We hereby note

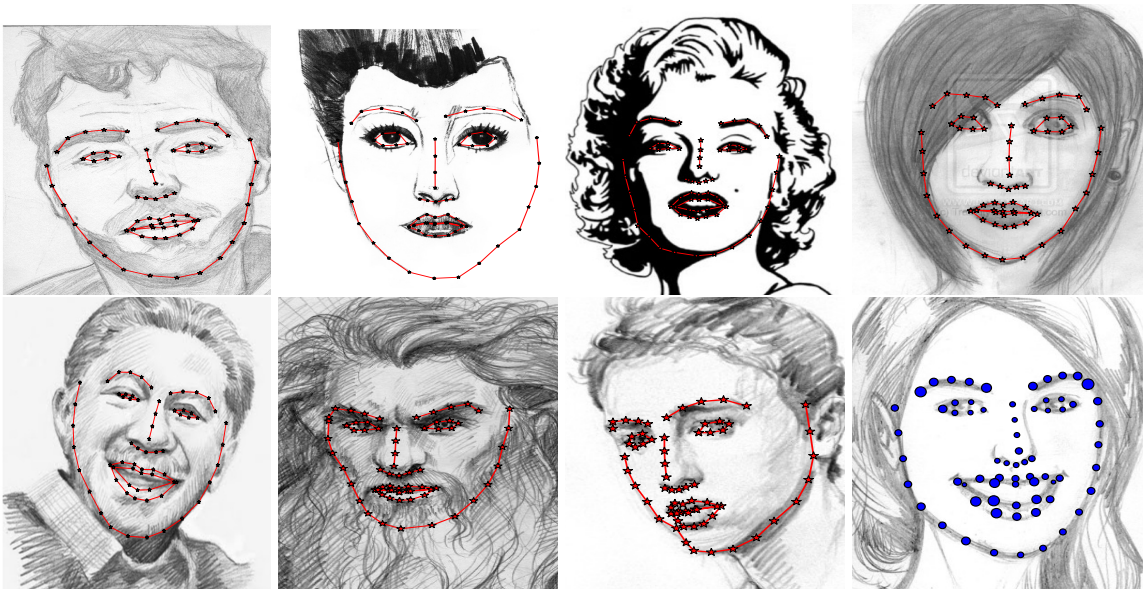


Fig. A.3 FSW example results. Face sketch images in FSW show large variety of head pose and drawing styles. The last image in the second row shows the average FSW individual landmark error levels, represented by the point sizes.

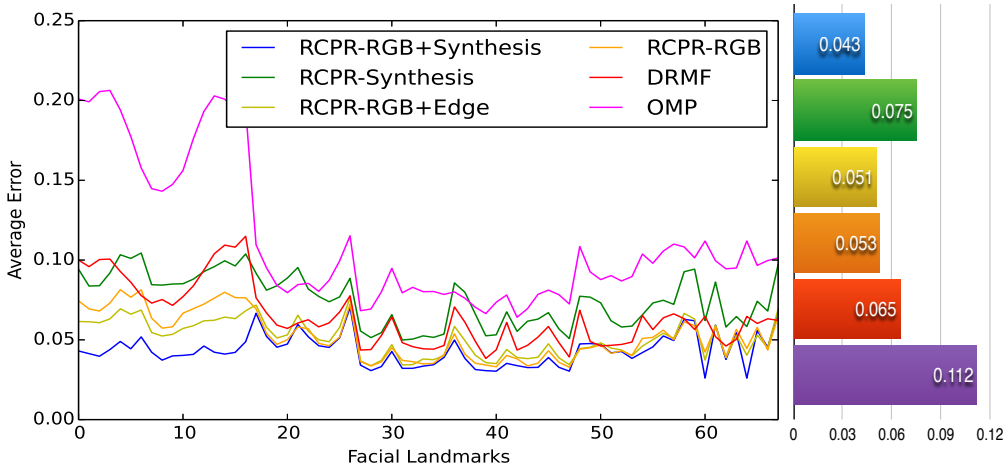


Fig. A.4 Results on the FSW dataset. The left shows the landmark-wise average error. The right shows the overall mean error. The landmark ID number definition please refer to (Sagonas et al., 2013c). Roughly, from #1 to #17 are landmarks along the face contour while the remaining are inner facial landmarks. For DRMF and OMP method, the inner mouth corners are not detected and their errors are shown as the mean value of all the landmarks.

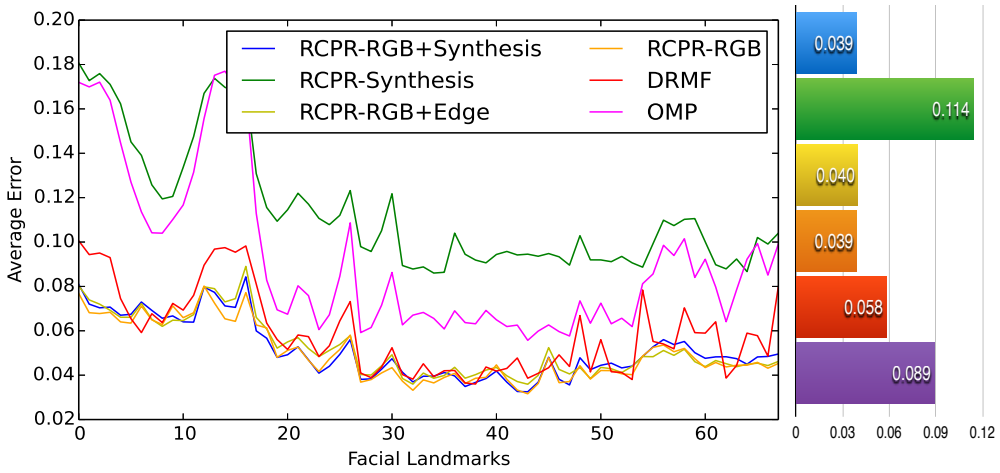


Fig. A.5 Results on the test images (RGB) of LFPW dataset. The figure configuration is the same to Fig. A.4.

that the number of training instances in Multi-PIE is much larger than that of the HELEN training set. The result is shown is in Fig. A.5. On RGB images, our proposed method performs on par with the other RCPR variants except the RCPR-Synthesis, which performed the worst on RGB images since it is only trained on synthesis images. All methods except RCPR-Synthesis perform better on RGB images than on the sketch images.

Though it is difficult for us to make exact comparison of the two modalities, we can observe our proposed method, and the DRMF method perform more consistently. However, the DRMF fails to achieve a high accuracy compared to our proposed method (6.5% vs. 4.3% on FSW and 6.8% vs. 3.9% on LFPW). For a conclusion, our proposed model, that is jointly trained on RGB images and their sketch synthesis, consistently performs better or very similar to the RCPR variants and the recent face landmark localization methods.